



Calhoun: The NPS Institutional Archive
DSpace Repository

Theses and Dissertations

1. Thesis and Dissertation Collection, all items

1986

NPS Supply Requisition Database - interactive software as an alternative to written instructions.

Trotter, Hartwell T.

<http://hdl.handle.net/10945/22003>

Downloaded from NPS Archive: Calhoun



Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

<http://www.nps.edu/library>

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93943

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

NPS SUPPLY REQUISITION DATABASE - INTERACTIVE SOFTWARE
AS AN ALTERNATIVE TO WRITTEN INSTRUCTIONS

by

Hartwell T. Trotter

March 1986

Thesis Advisor:

N. F. Schneidewind

Approved for public release; distribution is unlimited

T227299

REPORT DOCUMENTATION PAGE

a. REPORT SECURITY CLASSIFICATION			1b. RESTRICTIVE MARKINGS			
a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION/AVAILABILITY OF REPORT			
b. DECLASSIFICATION/DOWNGRADING SCHEDULE			Approved for public release; distribution is unlimited			
PERFORMING ORGANIZATION REPORT NUMBER(S)			5. MONITORING ORGANIZATION REPORT NUMBER(S)			
a. NAME OF PERFORMING ORGANIZATION		6b. OFFICE SYMBOL (If applicable)		7a. NAME OF MONITORING ORGANIZATION		
Naval Postgraduate School		54		Naval Postgraduate School		
c. ADDRESS (City, State, and ZIP Code)			7b. ADDRESS (City, State, and ZIP Code)			
Monterey, CA 93943-5000			Monterey, CA 93943-5000			
a. NAME OF FUNDING/SPONSORING ORGANIZATION		8b. OFFICE SYMBOL (If applicable)		9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER		
c. ADDRESS (City, State, and ZIP Code)			10. SOURCE OF FUNDING NUMBERS			
			PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
1. TITLE (Include Security Classification)						
NPS SUPPLY REQUISITION DATABASE - INTERACTIVE SOFTWARE AS AN ALTERNATIVE TO WRITTEN INSTRUCTIONS						
2. PERSONAL AUTHOR(S)						
TROTTER, Hartwell T.						
3a. TYPE OF REPORT		13b. TIME COVERED		14. DATE OF REPORT (Year, Month, Day)		15. PAGE COUNT
Master's Thesis		FROM _____ TO _____		1986 March		167
6. SUPPLEMENTARY NOTATION						
7. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)			
FIELD	GROUP	SUB-GROUP	INTERACTIVE SOFTWARE, REQUISITION DATABASE			
9. ABSTRACT (Continue on reverse if necessary and identify by block number)						
<p>Procedures to execute specific activities are usually communicated in writing throughout large organizations. This thesis presents a prototype example of interactive software as an alternative to the promulgation of written instructions. The Naval Postgraduate School supply requisition generation process has been distilled into a single software package, the Supply Requisition Database (SRdb), which prepares requisition documents and maintains a local database of items ordered. Emphasis is placed upon ease of use and labor efficiency. Although limited initial testing of the software is reported, SRdb is offered primarily as a tool for further research of the concept.</p>						
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT				21. ABSTRACT SECURITY CLASSIFICATION		
<input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS				UNCLASSIFIED		
22a. NAME OF RESPONSIBLE INDIVIDUAL			22b. TELEPHONE (Include Area Code)		22c. OFFICE SYMBOL	
N.F. SCHNEIDEWIND			(408) 646-2719		54Ss	

Approved for public release; distribution is unlimited.

NPS Supply Requisition Database -
Interactive Software as an Alternative to
Written Instructions

by

Hartwell T. Trotter
Lieutenant Commander, U.S. Navy
B.S., High Point College, 1971
M.A., Pepperdine University, 1976
M.S., University of Southern California, 1980

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
March 1986

ABSTRACT

Procedures to execute specific activities are usually communicated in writing throughout large organizations. This thesis presents a prototype example of interactive software as an alternative to the promulgation of written instructions. The Naval Postgraduate School supply requisition generation process has been distilled into a single software package, the Supply Requisition Database (SRdb), which prepares requisition documents and maintains a local database of items ordered. Emphasis is placed upon ease of use and labor efficiency. Although limited initial testing of the software is reported, SRdb is offered primarily as a tool for further research of the concept.

THESIS DISCLAIMER

The reader is cautioned that computer programs developed in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational and logic errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.

TABLE OF CONTENTS

I.	INTRODUCTION	8
A.	PURPOSE	8
B.	APPROACH	9
C.	ASSUMPTIONS AND SCOPE	9
II.	BACKGROUND	11
A.	NPS REQUISITION SYSTEM CHARACTERISTICS	11
B.	REQUISITION OVERVIEW	11
C.	TYPES OF REQUISITIONS	13
	1. Ready Supply Requests	13
	2. Standard Stock Requests	16
	3. Open Purchase	16
III.	SRdb SOFTWARE DESCRIPTION	21
A.	CONCEPT	21
B.	SYSTEM REQUIREMENTS	21
C.	FEATURES	22
	1. Document Generation	22
	2. Accounts Database	23
	3. Supplier Database	24
	4. Requisition Database	24
	5. Item Database	24
	6. Report Generation	25
	7. Alternative Sources Memorandum	27
IV.	INITIAL SRdb PROTOTYPE FEEDBACK	29
A.	GENERAL	29
B.	REQUISITION CODES	29
C.	DOCUMENT PRINTING	30
D.	DATABASE USAGE	31

V.	IMPLEMENTATION CONSIDERATIONS	32
A.	BENEFITS	32
B.	COSTS	33
C.	SRdb SCOPE	34
VI.	SUMMARY AND RECOMMENDATIONS	35
A.	FUTURE ADAPTATIONS OF SRdb	35
APPENDIX A:	ABBREVIATIONS AND ACRONYMS	37
APPENDIX B:	USER'S MANUAL	38
APPENDIX C:	DATABASE STRUCTURE	79
APPENDIX D:	DATA DICTIONARY	81
APPENDIX E:	SRdb SOURCE CODE LISTINGS	91
LIST OF REFERENCES	163
BIBLIOGRAPHY	164
INITIAL DISTRIBUTION LIST	165

LIST OF FIGURES

2.1	Typical Requisition Flow	12
2.2	Material Requirements Document	14
2.3	DD-1348 for Money Value Only	15
2.4	Office Supply Request Form	15
2.5	DD-1348 Standard Stock Requisition	16
2.6	DD-1348 Open Purchase Requisition (single item) . .	17
2.7	SF-36 Open Purchase Continuation	18
2.8	DD-1348 Open Purchase Requiring an SF-36	19
3.1	Single Requisition Summary	26
3.2	Department Listing of Items Not Received	26
3.3	Account Listing of Items Not Received	27
3.4	Alternative Sources Memorandum	28

I. INTRODUCTION

A. PURPOSE

Each day military commands collectively promulgate thousands of instructions, notices, and orders which subordinates are expected to read, understand, and execute. Although the authors of these written guidelines strive for clarity, the onus to understand them is placed upon the addressee. With the proliferation of personal computers the potential now exists to achieve new dimensions of clarity and ease of understanding through the issuance of interactive software vice voluminous written instructions.

Due to its capability to present a series of screens of information, software may be viewed as a communication medium. It is somewhat unique in that the user may interact with the software, for example, requesting further 'help' or clarification of particular procedures. The software also may be used to prepare reports by querying the user for inputs according to a specific logic. No pretention is made that software is the appropriate medium for the promulgation of all instructions. In fact, it is quite probable that it is cost efficient in only a small percentage of situations.

The purpose of this paper is to examine and propose a software-oriented alternative to the current manual, instruction-driven supply requisition process at the Naval Postgraduate School (NPS), Monterey, CA. Although the model presented addresses specific details of the NPS requisition transactions, it is felt that the general concepts could be extrapolated easily to other locations which also have significant open purchase activity and a multitude of funding sources.

B. APPROACH

The current NPS requisition process was studied in detail by reviewing the NPS Supply Department Customer Service Manual [Ref. 1] and the applicable NPS Comptroller instruction [Ref. 2]. Several modifications to these written guidelines were discussed at a two hour requisition training seminar presented by the Supply Department for NPS personnel [Ref. 3]. Finally, a working knowledge of the current system was attained through the interviewing of several persons involved in various stages of the requisition process at NPS. In addition to gaining a better understanding of the precise operation of the current system, suggestions for improvement were solicited.

Thesis research visits were made to the Fleet Hospital Support Office, Alameda, CA, Naval Supply Center, Oakland, CA, and the Supply Automation Office at NAS Miramar, CA, to review general supply automation efforts throughout the Navy.

C. ASSUMPTIONS AND SCOPE

The primary thrust of this paper is to present an example of interactive software as an alternative to a written instruction. Although cost effectiveness will be briefly examined as a relevant issue, a complete cost/benefit analysis is considered to be outside the scope of this thesis.

A requisition may be viewed as a compilation of data elements such as nomenclature, stock number, quantity, shipping codes, etc., which pertain to the item being ordered. During the course of research, some data elements required on requisition documents appeared to be a duplication of information, and other codes were discovered to be meaningless in the NPS environment. Since it is not the purpose of this thesis to present an analysis of information and codes required on each requisition document, these problems will be left to others.

For the purposes of this paper, every element of information and code which is required by applicable instructions and manuals [Refs. 1,2] are assumed to be necessary and appropriate. It is recognized that some of these requisition forms are in DOD-wide usage, and therefore, they may utilize codes which perhaps have meanings in other environments. An argument may easily be made that a locally produced requisition form would be more responsive to the needs of NPS supply customers, however, this is also felt to be an issue which is outside the realm of this thesis.

It is important to note that this paper focuses upon the accurate generation of requisition documents via a session with interactive software, as opposed to reliance solely upon written instructions. There is no intent to present an analysis of the entire NPS requisition / purchase process. For further information the reader is referred to the bibliography, which cites several previous theses which have fully discussed these other issues concerning the design and analysis of the NPS supply system.

II. BACKGROUND

A. NPS REQUISITION SYSTEM CHARACTERISTICS

During FY 85, the NPS Supply Department processed over 17,000 requisitions with a total value in excess of \$13,341,000. Due to the academic nature of its mission, the Naval Postgraduate School acquires a large percentage of its material via open purchase as opposed to reliance upon standard government stock sources. Less than 15% of FY 85 NPS requisitions were filled from standard stock, resulting in almost 15,000 open purchase actions valued at \$12,900,000 [Ref. 4].

NPS is somewhat unique in the number of different funding sources available for locally generated requisitions. Aside from the normal departmental Operating Target (OPTAR) funds provided in the NPS Operations and Maintenance, Navy (O&MN) budget, large amounts of Reimbursable Funds (RF) (also called research funds) are provided by others commands which are sponsoring research work at NPS. This wide variety of funding sources will be found to introduce a degree of complexity in accurate generation of requisition documents.

The current NPS Supply requisition process appears to be a manual anachronism in the midst of a relatively computerized environment. At present, no phase of the requisition cycle has been touched by automation. This chapter will describe the present procedures for generation of NPS requisition documents.

B. REQUISITION OVERVIEW

Each NPS department has designated one or more personnel to serve as the focal point in preparing and tracking requisition documents. For academic departments this person is

usually a clerk/typist serving in a GS-3 to GS-5 billet. Persons desiring to order materials usually notify the department clerk, who prepares the requisition documents and keeps a record of the transaction. The form of the initial request to the department clerk varies from one department to the next, running the gamut from a phone call, to a xerox copy of an advertisement with items circled, to a type-written memorandum.

The department requisition clerk prepares the documents in accordance with the applicable local instructions [Refs. 1,2]. After the documents are prepared, the clerk will obtain the appropriate signature prior to routing out of the department for further action. See Figure 2.1.

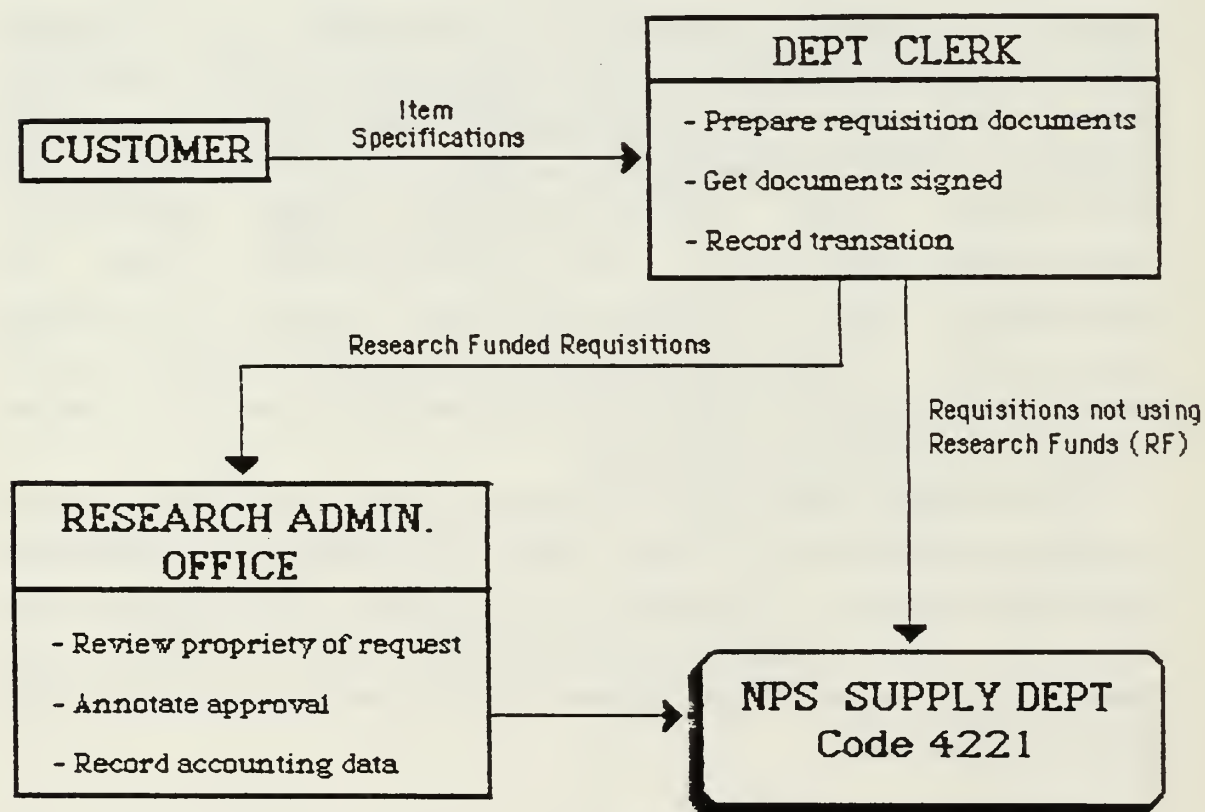


Figure 2.1 Typical Requisition Flow

If the requisition is to be funded from department OPTAR funding, the department chairman or his designated

representative will sign the documents before they are forwarded to supply. If the requisition is to be funded from Reimbursable Funds (RF), the documents will be signed by the professor controlling the specific RF account. RF-funded requisitions are sent to supply via the Research Administration Office which screens the request to ensure that the funds are being spent in accordance with any limitations placed by the sponsoring command.

C. TYPES OF REQUISITIONS

This section will describe the different types of requisitions and the associated documents.

1. Ready Supply Requests

To expeditiously satisfy customer requests, the most frequently requested consumable materials are stocked locally at the NPS Ready Supply Store (RSS). Materials stocked in the RSS for the convenience of the NPS customer are listed in the NPS RSS Catalog [Ref. 5]. Customers may obtain items from the RSS by forwarding a Material Requirements Document (Figure 2.2) to the RSS. RSS personnel will phone the exact total cost of the desired items back to the department requisitions clerk, who subsequently prepares a 'Money Value Only' DD-1348 (Figure 2.3) to be sent to the RSS.

The NPS Office Supply Issueroom (OSI) is a subunit of the RSS, where frequently used office materials, such as pens and folders, are available in limited quantities upon presentation of a locally produced office supply request form, Figure 2.4.

The OSI maintains a running tally of the value of items issued and 'bills' each department monthly. A department 'pays its bills' through issuance of a 'Money Value Only' DD-1348, as seen in Figure 2.3. The DD-1348 is an accounting document, and as such, it must be signed in the

Figure 2.2 Material Requirements Document

203J READY SUPPLY STORES										REQUISITION IS FROM (62271) NPS, AS Dept, Code 54 Rm 230 Bldg 330 X2472									
Z 9 S										MONEY VALUE ONLY									
X 62271 6022 3350 R 6 TED00										T 2D									
2 S 9Q OP 4 1 5										92.45									

Figure 2.3 DD-1348 for Money Value Only

OFFICE SUPPLIES; TURN-IN OR REQUEST
11 NO NPS 4400/1 (6-63)

FROM: Admin Science Dept, Code 54, ext 2472						DATE: 22 JAN 86		X		ISSUE	
TO: Supply Department, Office Supply Storeroom						CREDIT CARD NO.				TURN-IN	
APPROVED BY						DATE REQUIRED					
NO.	STOCK NO. AND DESCRIPTION					U / I	QTY	ACTION	U / P	COST	
1	STAPLER 7520-00-281-5895					EA	3		2.10	6.30	
2	PAPER, BOND 7530-00-290-0617					RM	4		2.60	10.40	
ISSUED BY						DATE		RECEIVED BY		TOTAL 16.70	

OFFICE COPY

Figure 2.4 Office Supply Request Form

upper right corner by an individual designated by the department as authorized to obligate funds [Ref. 1: p. 4-6].

2. Standard Stock Requests

Many commonly used items have been purchased in volume, assigned National Stock Numbers (NSN), and stocked at various stockpoints throughout the nation by the Department of Defense (DOD) or the General Services Administration (GSA). All standard stock items are requisitioned using a DD-1348, Figure 2.5, coded in accordance with the Military Standard Requisitioning and Issue Procedure (MILSTRIP) [Ref. 6: pp. 1-62]. A separate DD-1348 must be used for each line item of standard stock.

DOC IDENT										QUANTITY										REQUISITION IS FROM									
SENTO TO										(00228) NSC OAKLAND, CA										(62271) NPS, AS Dept, Code 54 Rm 230 Bldg 330 X2472									
A										TYPING STAND										A O A NOZ T									
C										X 62271 6022 3351 N 6 WGDMA A										T 2D									
FUND										DISTRIBUTION										PROJECT									
52 53										54 55 56										57 58 59									
2S										1W										OP4									
ADVANCE										A A										85.50 171.00									

Figure 2.5 DD-1348 Standard Stock Requisition

3. Open Purchase

Items which are not available from standard stock may be purchased directly from commercial sources. Although departments are issued standard stock catalogs annually, it is not uncommon for items, which are available from standard stock, to be erroneously requested on an open purchase requisition. All open purchase requisitions are screened by the NPS Supply Department for availability in standard stock. If the same or 'like' item is found, the requisition is returned to the originator with a memorandum attached citing the apparent standard stock substitution nomenclature

[illegible]

An open purchase requisition for a single item may be made using a DD-1348 as seen in Figure 2.6. A Standard Form 36 (SF-36) is required to be completed if more than one item is requested, or if the item nomenclature and description of specifications will not fit in blocks 8-22 on DD-1348 [Ref. 1: p. 4-5]. An SF-36 (Figure 2.7) is viewed as a continuation of the DD-1348 (Figure 2.8) in these instances.

For accounting purposes, items which are classified under different elements of expense must not be combined on a single requisition [Ref. 2: p. 3]. For example, an order for a personal computer and related software, even from a

STANDARD FORM 36, JULY 1966 GENERAL SERVICES ADMINISTRATION FED. PROC. REG. (41 CFR) 1-16.101		CONTINUATION SHEET		REF. NO. OF DOC. BEING CONT'D N62271/6030/5709		PAGE 1	OF 1
NAME OF OFFEROR OR CONTRACTOR GENERAL DYNAMICS 3234 CONVAIR AVE SAN DIEGO, CA 92129							
ITEM NO.	SUPPLIES / SERVICES	QUANTITY	UNIT	UNIT PRICE	AMOUNT		
1	20Mbyte Tape Streamer Model # 123456 Version 9.3 Black Finish	3	EA	275.00	825.00		
2	50Mbyte Hard Disk (IBM PC Compatible) 110 volts 5" X 9" X 2"	3	EA	500.00	1500.00		
					TOTAL		
					2325.00		

Figure 2.7 SF-36 Open Purchase Continuation

DOC IDENT		EQUIP IDENT		FISC		WITH		200		QUANTITY		REQUISITION		VARI		SERIAL		SUPPLEMENTARY		DATE		PROJECT		FISC		DATE		STATE			
STOCK NUMBER		STOCK NUMBER		STOCK NUMBER		STOCK NUMBER		STOCK NUMBER		STOCK NUMBER		STOCK NUMBER		STOCK NUMBER		STOCK NUMBER		STOCK NUMBER		STOCK NUMBER		STOCK NUMBER		STOCK NUMBER		STOCK NUMBER		STOCK NUMBER		STOCK NUMBER	
SEND TO GENERAL DYNAMICS 3234 CONVAIR AVE SAN DIEGO, CA 92129																REQUISITION IS FROM (62271) NPS, AS Dept, Code 54TH Rm 230 Bldg 330 X2472															
ROUTING DATA DOC IDENT: 1 2 3 4 5 6 7 ROUTING IDENT: 1 2 3 4 5 6 7																STOCK NUMBER See Attached SF-36															
DOCUMENT NUMBER X 62271 6030 5709 N 6 RHUTE A																REMARKS W 2D															
FUND 2S																STATUS DATA 2325.00															
ADVISE AA																2325.00															

Figure 2.8 DD-1348 Open Purchase Requiring an SF-36

single vendor, requires at least two requisitions because the expense element codes for software and hardware are different [Ref. 2: encl(3) pp. 1-2].

a. Alternative Sources Memorandum

Recently publicized purchases of '\$600 ashtrays' and '\$900 hammers' have focused intense congressional scrutiny upon DOD procurement. In an attempt to ensure efficient use of tax dollars through competitive purchases, all open purchase requisitions with a total value exceeding \$1,000 are now required to list a minimum of three alternative vendors for the requested materials. The alternative suppliers may be shown in the space remaining at the bottom of an SF-36, or a separate memorandum may be attached to the requisition [Ref. 3].

b. Sole Source Justification Memorandum

An exception to the Alternative Sources Memorandum occurs when the customer is able to document the requirement for a sole source procurement. Contracting officers are 'bound by law to seek competition' unless it can be demonstrated that valid reasons exist which preclude competition; therefore, requests for sole source procurement

actions are very carefully reviewed [Ref. 1: p. 4-27]. Central to the Sole Source Justification is a description of the critical features which limit its availability to a single source, as well as a certification that this is the only product or service which will adequately meet the intended use or application.

III. SRdb SOFTWARE DESCRIPTION

A. CONCEPT

The Supply Requisition Database (SRdb) software was written to permit the generation of primary requisition documents through an interactive computer session. The target user, a department requisition clerk, initiates the order process by answering a logical series of plain language queries. Since the algorithms for placing the appropriate codes in the proper boxes on the documents are written into the software, the user is freed from memorizing or copying cryptic codes. After all required questions have been answered, the user is prompted to load the proper blank document into the printer. Precise document formatting is provided in the SRdb software, eliminating all user format errors.

The SRdb program is designed to extract key elements from each document and maintain a requisition database which can be used at the department level in lieu of the current ad hoc recordkeeping. SRdb files are written to the selected disk drive as per the SRdb configuration file. The configuration file is used to maximize the user friendly nature of the software and free the user from repetitive typing of elements which remain constant on each requisition, such as department name, phone, building, etc. The beta (initial test) copy of the SRdb package was preconfigured for use in the NPS Administrative Sciences department.

B. SYSTEM REQUIREMENTS

SRdb is designed to run on an IBM or IBM compatible CPU with a minimum of 256K of memory and 2 disk drives. Although it will operate small databases on floppy disk drives, it is highly recommended that a hard disk drive be

utilized if any of the following requirements are exceeded on an FY basis:

- 50 accounts,
- 150 suppliers,
- 1500 requisitions, or
- 2000 line items.

Either a color or a monochrome monitor is acceptable. A friction feed printer is essential for printing of forms, however, a tractor feed printer is adequate for printing out reports and memorandums.

Since SRdb is essentially a dBASE III application program, a copy of dBASE III is required for program execution. The beta copy of SRdb is configured such that dBASE III and the SRdb program modules are resident on a single disk to be maintained in the 'A' disk drive. The database and index files are located on the 'B' drive. These drive assignments are optional and may easily be changed by modifying the configuration file. For further details concerning system sizing requirements, the reader is referred to Appendix C, 'Database Structure'.

C. FEATURES

The general features of the SRdb software are described below. The reader is referred to the SRdb User's Manual in Appendix B for detailed operational descriptions.

1. Document Generation

The two primary requisition documents, DD-1348 and SF-36, are readily generated by SRdb. SRdb handles complex code assignment in document boilerplate according to the type of requisition. Julian dates are automatically calculated, including leap year corrections. It also calculates the total value of the requisition required on the SF-36 and in block 'U' of the DD-1348 [Ref. 1: pp. 4-9 - 4-15].

When ready to print, the program pauses for the user to indicate that the proper form has been loaded into the printer. After the document has been printed, SRdb permits the user to immediately reprint it in case the printer was misloaded, or there were any mechanical malfunctions. Although both forms are available on continuous-feed stock, it is believed that it generally would be impractical to use such forms due to the variety of requisition types. For example, each SF-36 would necessarily be followed by a DD-1348 as described in Chapter Two, therefore, a friction feed printer is viewed as most practical. A single SRdb-generated DD-1348 or SF-36 document would be indistinguishable from the same document prepared by a clerk-typist on an electric typewriter, except perhaps by the lack of overstrike errors.

2. Accounts Database

SRdb allows the clerk to select the appropriate account from an on-line listing, eliminating the repetitive entry of account information. Due to the proliferation of RF projects throughout NPS, it is not uncommon for a single department to have access to 30 or more different funding accounts. SRdb records the following information for each account:

- Account name,
- Account number assigned by NPS Comptroller,
- Mail code of responsible individual,
- Type of funds (RF or not),
- Document serial number range assigned,
- Last document serial number used.

Professors who are responsible for multiple RF accounts have a separate listing for each account. A menu-driven option permits the user to easily add additional accounts upon receipt of accounting data from the NPS Comptroller.

Tracking document serial numbers for each account permits the automatic generation of the 14-character document numbers (columns 30-43 of DD-1348), which are a concatenation of the Unit Identification Code (UIC), Julian date, and the appropriate serial number. In SRdb, the entire process is now transparent to the user instead of a being a drill in precision typing.

3. Supplier Database

A database file of suppliers is maintained on-line to assist the user in completion of open purchase requisitions. The vendor's name, address, and phone number are filed for easy future reference. SRdb allows the clerk to select the appropriate supplier from an on-line listing, eliminating the repetitive entry of vendor information. If an order is made from a supplier not currently in the file, he is automatically added to the database for future reference. The supplier database also serves as an excellent source of alternative vendors, required in orders exceeding \$1,000 value.

4. Requisition Database

A file is maintained containing the following items of information about each requisition:

- Requisition number,
- Priority code of requisition,
- Supplier,
- Date of requisition.

The requisition database is indexed by requisition number and account number for ease of report generation. Each requisition is represented by a separate entry in this file.

5. Item Database

In consonance with a database of Third Normal Form, as described by Kent [Ref. 7], a separate database contains

the items requisitioned. The database contains the following entries for each item:

- Description of item (up to 44 characters),
- Unit of issue,
- Quantity ordered,
- Unit price,
- Requisition number,
- Purchase order number,
- Receipt status.

The item database is indexed by both the requisition number and the purchase order number to facilitate item tracking and report production. This database also permits the tracking of which items have not been received. The original unit price may later be altered to accommodate price changes, which are a common occurrence.

6. Report Generation

A primary strength of SRdb is the ability to easily generate reports from the various databases. The user is given the option of displaying each report on the screen or dumping it to a printer. Although there is a virtually unlimited number reports and formats which could be produced, the beta version of SRdb provides the following two basic types of reports.

a. Requisition Summary

The user may select from these three requisition display options:

- 1) A single specific requisition.
- 2) All requisitions charged to a specific account.
- 3) All requisitions charged to a specific account between two dates.

As may be seen in Figure 3.1, only essential data elements are presented in requisition summaries. There is no attempt to display an exact copy of the original requisition.

Account Name: LAPATRA
Account #: RGAZ2

Requisition #: 53337003
Priority: C

To: ASHTON-TATE
10150 W. Jefferson
Culver City, CA 90230

#	Item	Unit Cost	Quant	Rcvd	P.O. #
1	dBASE III	\$ 399.00	2	F	
2	Quick Code Mark IV	\$ 345.00	1	F	87654321

Total items: 3 Total value: \$ 1,143.00

Figure 3.1 Single Requisition Summary

b. Summary of Items Not Received

SRdb currently permits two types of displays of items ordered but not yet received. The user may elect to display all items outstanding for the entire department (Figure 3.2) or to restrict the list to items outstanding from a specific account (Figure 3.3). Both types of listings will show at the bottom the total number and value of items not yet received.

AS Dept

ITEMS NOT RECEIVED AS OF 12/11/85

#	Item	Req #	P.O. #	Quant
1	LOTUS 123 34LP45654323	53156504		1
2	Pedometer (model 158)	53227676		2
3	Nikon 35mm SLR Camera fl.4	53227676	53403454	1
4	Calculator 34543LP09867	53306005		1
5	DisplayComm Software	53336505	54129990	1
6	IBM VM-370 Oper. Handbook	53336505		1
7	dBASE III	53337003		2
8	Quick Code Mark IV	53337003		1
9	Amdek Color Monitor 13'	53337678		2
10	9 Meg Buffer Board	53365707		2
11	Calender 86 78756GG32234	53367005		3
12	Desk Pad 66543FU45435	53367702		1
13	Symphony Clone Vers #3	53385708		3
14	Sidekick Software	53387680		100
15	IBM AT Personal Comp.	53387703		1
16	Y-19 Communications Accessory Pack	53415709		1

Total items: 182 Total value: \$ 64,656.62

Figure 3.2 Department Listing of Items Not Received

The potential exists to restrict the listings to requisitions generated within a specific range of dates.

For Account #: RCAZ2

ITEMS NOT RECEIVED AS OF 01/23/86

#	Item	Req #	P.O. #	Quant
1	dBASE III	53337003		2
2	Quick Code Mark IV	53337003		1
3	Calender 86 78756GG32234	53367005		3

Total items: 6

Total value: \$ 1,151.61

Figure 3.3 Account Listing of Items Not Received

7. Alternative Sources Memorandum

As described in Chapter Two, a memorandum listing three alternative suppliers must accompany all open purchase requisitions exceeding \$1,000 in value. SRdb permits the user to page through the on-line list of available suppliers, selecting the desired ones. The user is then prompted for two departmental points of contact for the requisition. The resulting memorandum (Figure 3.4) is produced without further user input.

From: AS Department

01/24/86

To: NPS Supply

Subj: Additional sources for requisition # 60243397

1. Due to the high value of subject requisition, the following multiple supply sources are submitted:

Monterey Bay Computerworks	Phone: 408 889-3177
1760 Fremont Blvd	
Seaside CA	93955

IBM Government Products	Phone: 518 864-2169
321 Cypress Lane	
Oakmont NM	43563

Trafalgar Systems	Phone: 808 549-2843
606 Alamoana Blvd	
Honolulu, HI	96867

2. Departmental points of contact for this request are:

Greta Jones	2472
-------------	------

Bette Midler	3242
--------------	------

Figure 3.4 Alternative Sources Memorandum

IV. INITIAL SRdb PROTOTYPE FEEDBACK

This chapter will present a brief overview of the lessons learned from the limited trial operations of the beta copy of SRdb in the NPS Administrative Sciences Department. No pretention is made that these findings constitute a formal cost/benefit analysis of the concept, however, it is believed that they may serve a springboard for further research in the area. Feedback was obtained by direct observation of the requisition clerk users and post-use interviews.

A. GENERAL

The first trial of SRdb in the AS Department was conducted using a newly hired clerk who was unfamiliar with both the requisition process and personal computers. Her experience highlighted several procedural weaknesses in the software which were immediately corrected prior to further use.

Users with all levels of experience in the requisition arena appeared generally favorable toward the SRdb concept. The best reception was from those with 0 to 5 years of requisition experience. Although the user's enthusiasm for the SRdb concept appeared genuine, it is recognized that their knowledge of the presence of the software's author may have somewhat colored their reactions. All said that it was superior to the current system of learning, which consists primarily of literally copying previous requisitions, substituting the name, number, and price of the new item to be ordered.

B. REQUISITION CODES

The primary problem with the verbatim copying of previous requisition stubs is the inadvertent transcription

of inappropriate codes. Users have always found it easier to copy a sample, rather than research and understand the actual meanings of the codes. For example, after the issuance of the current NPS Customer Service Manual, the Supply Department noticed that the expense code 'T 2D' was being placed in block 'O' of virtually all DD-1348's. The cause was traced to the fact that all sample DD-1348's shown in the new manual used the 'T 2D' code as an example [Ref. 1: pp. 4-10 - 4-14]. Most requisition clerks were found to be blindly copying it, not really understanding why, but thinking that it should work [Ref. 8].

A valid argument may be made that it is unnecessary for clerks to understand the meaning of every single code on requisition documents. Indeed, six of the 22 codes found on a DD-1348 will never change for an NPS customer, therefore, it is acceptable for clerks to routinely copy those items. However, when a code is variable, under the current system the clerk should understand its significance.

Due to the internalization of code algorithms within the SRdb software, it was most favorably received by inexperienced personnel who had not yet learned the details of the requisition coding process. SRdb generates the appropriate codes based upon user response to plain language queries, thereby relieving the user from memory work or copying exercises.

C. DOCUMENT PRINTING

The actual printing of requisition documents by SRdb was initially viewed by all users with concern. Due to its small size, the DD-1348 requires precise character placement, and no one was comfortable with alignment of the forms in the printer. It is a process which one learns through trial and error. Since SRdb permits an unlimited number of immediate reprinting attempts, the skill is rather easily acquired. No one required more than four trials to print his or her first

requisition. A rapid learning curve for subsequent uses was discovered for nearly all users.

D. DATABASE USAGE

The notion of reliance upon an electronic database rather than the current '3-ring binder' system produced surprisingly mixed reviews. It was generally felt that unless the SRdb system was already running at the time a single requisition needed to be reviewed, it might be easier and faster to locate it in the current binder system. All agreed that SRdb was superior in the generation of summary reports of items not yet received, however, this was not viewed as a feature which would be in frequent demand in the AS Department. Everyone liked the concept of maintaining on-line files of suppliers and account data to avoid typing of redundant information on multiple requisitions.

SRdb was judged by some users to be weak in that the beta version requires all requisitions for a particular account to be generated via SRdb. In other words, it is difficult to 'juggle' the document serial numbers if a particular requisition is manually produced on a typewriter. This weakness could be overcome in two ways. An additional module could be created allowing the user to change the serial number of the next document, however, this has serious implications in terms of the ultimate integrity of the database. Alternatively, a department policy could be adopted requiring all requisitions to be generated by SRdb, thereby eliminating the possibility of conflicting manually generated document serial numbers.

V. IMPLEMENTATION CONSIDERATIONS

A. BENEFITS

In the case of SRdb, the following potential benefits may be derived from the issuance of interactive software instead of written instructions and manuals:

1. Requisitions may be generated more quickly and easily, requiring less effort and perhaps permitting a reduction in user labor expense.
2. The time required to train users to produce requisition documents may be significantly reduced.
3. SRdb permits the automatic generation of various department level requisition summary reports which may be of value to management, however are not now produced due to their labor intensity in a manual system.
4. A consistently higher quality of requisition document may assist the NPS Supply Department in providing better response to customers. Many common requisition coding and procedural errors would be eliminated, thereby, reducing the number of documents returned to users for correction.
5. SRdb's automatic generation of requisitions through a series of, progressive screens may eliminate many of the user's telephone queries concerning procedural matters to the NPS Supply Department personnel, saving time for both.

Since only the last two of the five potential benefits would be enjoyed by the NPS Supply Department, it is legitimate to question whether it would be sufficiently in their interest to sponsor the implementation of an SRdb project. Although a majority of the benefits will be realized by other NPS departments, the bulk on the development and maintenance expenses probably would fall upon the Supply Department. This would call for a change in the typical institutional inertia, deeply entrenched in division of labor and turf considerations, which notes that 'we will tell them how to write a requisition, and we will process their finished document, but we are not going to write it for them.' In other words, despite the net potential benefit to the command as a whole, it would be natural for the

Supply Department to continue to publish written instructions, and let the customers in the other departments worry about reading, understanding, and executing them. Facing such natural barriers, if the Supply Department does not view benefits 4 and 5 as sufficient to warrant the expense from their viewpoint, it is probable that the concept represented by SRdb will require sponsorship from someone at a sufficiently high echelon within the command to be concerned with the overall common benefit.

B. COSTS

The current beta copy of SRdb required 450 hours of programming efforts by a relatively novice programmer. After software development, the maintenance phase continues to represent a substantial investment, frequently exceeding 50% of the total cost over the program's life-cycle [Ref. 9]. As requisition procedures are modified in the future, new versions of SRdb will require code alterations. SRdb may offer an advantage in that many modifications could remain transparent to the user, however, the costs of program changes must still be paid, probably by the originator. The concept would be for new versions of floppy diskettes to be issued in lieu of the promulgation of change notices to written instructions.

The disparity between media costs (diskette versus paper and ink) may not be as great as one might imagine. Now that diskettes may be procured in bulk for less than \$1.00 per copy, the media costs are comparable and are considered to be a primary consideration.

Viewed from a system perspective, SRdb requires the use of substantial assets beyond the cost of the SRdb software. Command-wide implementation of SRdb would of necessity force a review of the availability and utilization levels of personal computers throughout the command. NPS is somewhat unique in that the proliferation of personal computers has

already enabled access to them in every department, however, it would certainly be a major consideration in a more diverse environment.

C. SRdb SCOPE

It should be recognized that SRdb extends well beyond the bounds required to provide a software implementation of the requisition process as currently described in the appropriate written references [Refs. 1,2,6]. The incorporation of a database capability in terms of a formal record keeping structure exceeds actions currently directed. The database features were included as a result of a survey of the record procedures of various departments.

One may question the wisdom of stipulating the form of department records when there has been no guidance previously given in this area. While no advantage has been conclusively proven supporting a requisition record system that is uniform command-wide, there is an appealing logic to having one group create a system which is automatically updated at the time of requisition generation with no additional effort by the user. This is as opposed to the *laisse faire* process of each department reinventing the 'record wheel' separately. Certainly a department could use SRdb for requisition preparation, choosing to ignore the SRdb-produced records, and maintaining its manual binder file system instead. In all likelihood, if a descendent of SRdb is ever formally implemented, the manual records will continue to be kept in parallel with SRdb until user confidence is fully gained.

VI. SUMMARY AND RECOMMENDATIONS

SRdb is presented as an example of interactive computer software as an alternative to written instructions. The concept, as embodied in the SRdb prototype, has been informally tested and successfully demonstrated to work in a closely controlled environment. Despite the appearance of high potential and the numerous common sense arguments which may be advanced in favor of software over written instructions, it is inappropriate to conclusively state that SRdb offers proof of cost efficiency or effectiveness of the concept. SRdb is offered as a tool for further research and analysis in this arena.

A. FUTURE ADAPTATIONS OF SRdb

SRdb may be viewed as an attempt to improve the human interface with the NPS supply requisition process without modifying the basic system. Although document generation is now automated by SRdb, the physical transmission of requisition data on paper stubs remains far less efficient and somewhat archaic when compared to other procedures which are readily available today. For example, an SRdb-like interface could easily electronically send the requisition to a host computer in the Supply Department, rather than produce paper stock which is manually routed. Supply procurement personnel could receive, review, and process requisitions on their terminals, eliminating enormous amounts of duplication in typing efforts. A 'read only' requisition supply status database could also be maintained on the NPS mainframe, accessible by modem or 3278 terminal from every department, eliminating substantial hours in answering routine status check phone calls.

Although current supply automation experts believe that the Navy is not yet ready for completely 'signatureless' electronic requisitions from the customer level, as a compromise, the system could permit automatic electronic transmission and processing of all requisition data elements to be followed by a simple signed memorandum citing each requisition [Refs. 10,11].

Organizations are just beginning to scratch the surface in realizing the potential of personal computers when teamed with the appropriate software. SRdb is a minor contribution to this abrasion process.

APPENDIX A
ABBREVIATIONS AND ACRONYMS

AS	Administrative Sciences
Beta	Initial Test Copy of Software
Char	Character
DBF	Database File
DOD	Department of Defense
FY	Fiscal Year
GSA	General Services Administration
MILSTRIP	Military Standard Requisition and Issue Procedure
NDX	Index File
NPS	Naval Postgraduate School
NSC	Naval Supply Center
NSN	National Stock Number
O & MN	Operations and Maintenance, Navy
OPN	Other Procurement, Navy
OPTAR	Operating Target
OSI	Office Supply Issueroom
RF	Reimbursable Funds
RSS	Ready Supply Store
SRdb	Supply Requisition Database
UIC	Unit Identification Code

APPENDIX B

SRdb USER'S MANUAL

Table of Contents

I.	TO THE NEW USER	40
II.	START UP PROCEDURES	41
III.	CONFIGURATION	43
IV.	DATE CONFIRMATION	45
V.	THE MAIN MENU	46
VI.	COMMONLY USED PROCEDURES	47
	A. ACCOUNT SELECTION	47
	B. SUPPLIER SELECTION	49
	C. REQUIRED DELIVERY DATE	51
	D. ORDER PRIORITY	52
	E. PREPARE PRINTER	53
VII.	TO PLACE AN ORDER	55
	A. OPEN PURCHASE	56
	1. Enter Item to be Ordered	56
	B. STANDARD STOCK REQUISITION	58
	1. Select Standard Stock Source	58
	2. Enter Item to be Ordered	58
	C. NSC OAKLAND FORMS REQUISITION	59
	D. PUBLICATION REQUISITION	60
	E. RSS REIMBURSEMENT	61
	F. TRANSPORTATION OF MATERIAL REQUISITION	62
VIII.	UPDATE FILES WITH PURCHASE ORDER	64
IX.	MARK ITEMS RECEIVED	66
X.	LIST ITEMS NOT YET RECEIVED	68
XI.	DISPLAY A REQUISITION	70
XII.	CHANGE AN ITEM PRICE	72
XIII.	ENTER A NEW SUPPLIER	75
XIV.	CREATE A NEW ACCOUNT	77

LIST OF SCREENS

2.1	Welcome	41
2.2	Load Disk Drives	42
3.1	Configuration Banner	43
3.2	Configuration Modification	44
4.1	Confirm date	45
4.2	Enter Date	45
5.1	Main Menu	46
6.1	Select an Account	47
6.2	Selecting by Account Name	48
6.3	Scrolling to the Previous Account	48
6.4	Selecting an Account for Use	49
6.5	Select a Supplier	49
6.6	Looking for IBM	50
6.7	Selecting IBM	50
6.8	Required Delivery Date	51
6.9	Order Priority	52
6.10	Priority 'A' Warning	52
6.11	Prepare Printer Warning	53
6.12	Device Error	53
6.13	Load SF-36	54
6.14	Load DD-1348	54
7.1	To Place an Order	55
7.2	Open Order	56
7.3	Sample Open Order	57
7.4	Standard Stock Sources	58
7.5	Standard Stock Item Entry	59
7.6	Forms Order	60
7.7	Publication Order	61
7.8	RSS Reimbursement	61
7.9	Transportation of Material	62
7.10	Prepare to Enter Transportation Number	63
7.11	Enter Transportation Serial Number	63
8.1	Input P.O. #	64
8.2	Input Req # and PO #	64
8.3	Requisition Not Found	65
8.4	Item Displayed for PO # Input	65
9.1	Mark Receipt of an Item	66
9.2	Pick Requisition Number or PO #	67
9.3	Item Not Shown as Received	67
10.1	List Items Not Received	68
10.2	Search Type Selection	69
11.1	Display Requisitions	70
11.2	Display Requisition Options	71
11.3	Requisition Generation Dates	71
12.1	Change Item Price	72
12.2	Select Item by Req # or PO #	73
12.3	Enter Requisition Number	73
12.4	Requisition Not Found	73
12.5	Item Displayed for Price Change	74
13.1	Enter New Supplier	75
13.2	Input new Supplier	76
13.3	New Supplier Example	76
14.1	Enter New Account	77
14.2	Input New Account Screen	78
14.3	New Account Example	78

I. TO THE NEW USER

The Supply Requisition Database (SRdb) program is designed to be easily used by a person with very little knowledge of the NPS Supply system. The menu-driven SRdb approach is best learned through experience. The purpose of this User's Manual is to serve as a reference guide if more details about a particular process are required.

Assuming that your copy of SRdb has been configured for your department, there is no need to read further. Simply follow your department's instruction for disk placement and start up SRdb. It is recommended that new users use a test database diskette for the first session, so that practice requisitions and reports may be generated with no penalties for mistakes. Enjoy....

II. START UP PROCEDURES

WELCOME TO THE NPS SUPPLY REQUISITION DATABASE

This program is designed to originate requisition documents (DD-1348s and SF-36s), as well as maintain a database of requisition status for a particular department. The system is designed for use by any personnel who are familiar with Naval Supply terminology, however, a detailed understanding of the decision matrix used in the generation of requisition documents is unnecessary.

WARNING - If the user fails to adhere to any warnings presented by the program, he does so at the risk of data loss.

Disks should NEVER be removed from the drives until the program has been exited.

Screen 2.1 Welcome

The SRdb program is presented through a series of menus which are felt to be self-explanatory. This user's guide will present copies of actual screen displays. For example, Screen 2.1 is the first seen when the SRdb program is started. The space provided for user inputs would be seen as a cursor on the screen. In example screens in this manual, the cursor will be represented by an underline . If the screen is demonstrating an example of the choice made by a user, it will show the selection made as an underline, e.g. Y would represent a user's input of 'Y'.

If this is the first time the program is being run in your department, it should be configured by someone familiar with dBASE III and the Program Maintenance Manual.

1. Turn on the computer & boot up DOS. If you don't know how to do this, get someone to do it for you, or read your computer instruction manual.
2. Assuming the program is configured for use in your department, load the program in accordance with the department's instructions. If none exist, then the program disk should be placed in the 'A' drive and the database disk placed in the 'B' drive.
3. Type 'ORDER'.
4. Follow instructions to 'Press a key to continue' until you see the Screen 2.2

LOAD DISK DRIVES

Database Disk Location

Suppliers: 'B' Drive

Accounts: 'B' Drive

Screen 2.2 Load Disk Drives

Ensure that the disk drives are properly loaded before proceeding! If you have loaded the disks in locations other than the locations now on the screen, this is the time to change them.

III. CONFIGURATION

CONFIGURATION. -----

This program has been preconfigured for your use and should not require reconfiguration unless transported to another system or department.

WARNING - Configuration changes should only be made by personnel familiar with dBASE III.

Change Configuration (Y/N)?

Screen 3.1 Configuration Banner

Normally, the proper response to Screen 3.1 should be 'N'. The beta copy of SRdb has been preconfigured for the NPS Administrative Sciences Department. If you respond with a 'Y' to Screen 3.1, you will be given the opportunity to modify the default values shown on Screen 3.2. This process should be necessary only once per department.

This program is configured as follows:
Department: AS
Bldg: 330
Room: 230
Telephone: X2472
Database disk locations:
Suppliers: 'B' drive
Accounts: 'B' drive
Color Monitor (T/F): T
Is this the correct configuration (Y/N)? __

Screen 3.2 Configuration Modification

Responding to Screen 3.2 with a 'Y' will cause the displayed configuration to be saved to the program disk.

IV. DATE CONFIRMATION

Since the computer's system date may not have been correctly set, the opportunity is now presented to enter the desired date. This date will be used in dating all requisition documents and reports, therefore, it is important to ensure that it is correct.

CONFIRM TODAY'S DATE

12/11/85

(Y/N)?

Screen 4.1 Confirm date

If the date shown in Screen 4.1 is correct, enter 'Y'. If you enter 'N', Screen 4.2 will allow you to change the date as you desire.

ENTER TODAY'S DATE

MM/DD/YR

Note - Leading zeros are required
for single digit values.

Screen 4.2 Enter Date

V. THE MAIN MENU

```
=====
=                                     =
=      SUPPLY REQUISITION DATABASE  =
=                                     =
=====
=                                     =
=      0.  Exit                      =
=      1.  Place an order           =
=      2.  Update files with purchase order # =
=      3.  Mark items received      =
=      4.  List items not yet received =
=      5.  Display a specific requisition =
=      6.  Change an item price     =
=      7.  Enter a new supplier     =
=      8.  Create a new account     =
=                                     =
=====Your choice >_<=====
```

Screen 5.1 Main Menu

Screen 5.1 is a display of the SRdb Main Menu. The following is a description of the various menu options.

- 0) Quit and exit the program.
- 1) Place an order. Generates the appropriate requisition documents and places the order in the database.
- 2) Permits the entry of a purchase order number for each item when the purchase order information is received from supply.
- 3) Permits the marking of items which have been received.
- 4) Generates a listing of items which have not been received.
- 5) Displays a specific single requisition or a series of requisitions based upon a particular requisition number or account number. The requisitions displayed may be made subject to a range of dates of origin.
- 6) Permits the price of an item to be changed in the database. It does not regenerate requisition documents as this is generally unnecessary.
- 7) Allows a new supplier to be entered into the database.
- 8) Permits the creation of a new account when the appropriate information is received from the NPS Comptroller.

VI. COMMONLY USED PROCEDURES

There are several procedures which are used many times throughout the SRdb program. This chapter will discuss all procedures which are common to two or more major routines. For example, the procedures to select an account are the same for all six different types of requisitions, therefore, account selection is a procedure which will be discussed here.

A. ACCOUNT SELECTION

SELECT AN ACCOUNT		

-	1. By Account Name	-
-	2. By Account Number	-
-	3. Enter New Account	-
-	4. Return to Main Menu	-

Your choice: <u>1</u>		

Screen 6.1 Select an Account

Many routines require the user to select a particular account for further activity. The desired account may be specified by either account name or account number. If the account is new, the option is given to create a new account. Since creation of a new account is also an option on the main menu, it will be discussed later. For purposes of the example at hand, it is assumed that the desired account is already on file, and the user chooses to specify it by name, yielding Screen 6.2

SRdb displays data on the 'LYONS' account in response to Screen 6.2. For user convenience, the name input is not sensitive to upper or lowercase. If no account having the desired name can be located in the database, the user is so informed and instructed to try again. The dBASE III search procedure allows for input of partial names, such as 'LY' instead of 'LYONS'.

Once the account is found, the user has several options as shown at the bottom of Screen 6.3. For purposes of this example, we have chosen to page back to the previous account in the database (files are alphabetically arranged by account name). This brings up the 'LAPATRA' account as seen in Screen 6.4.

Account Name: LYONS
(Blank aborts process)

```

Account Name:  LYONS
Account Number:  R1235
Code:  54LY
Research Account (T/F):  T
Requisition Serial # Range -  From:  6500
                               To:    6505

      Select an action: 2
-----
-  1. Use this account  -
-  2. Scroll to previous account  -
-  3. Scroll to next account  -
-  4. Return to last menu  -
-  5. Return to main menu  -
-----

```

The LAPATRA account displayed in Screen 6.4 is selected for use by choosing option '1' as shown.

Account Name: LAPATRA
Account Number: RCAZ2
Code: 54LP
Research Account (T/F): T
Requisition Serial # Range - From: 7001
To: 7005

Select an action: 1

- 1. Use this account -
- 2. Scroll to previous account -
- 3. Scroll to next account -
- 4. Return to last menu -
- 5. Return to main menu -

Screen 6.4 Selecting an Account for Use

B. SUPPLIER SELECTION

SELECT A SUPPLIER

- 1. By Supplier Name -
- 2. By Supplier Number -
- 3. Enter New Supplier -
- 4. Return to Main Menu -

Your choice: 1

Screen 6.5 Select a Supplier

Many routines require the user to select a particular supplier for further activity. The desired account may be specified by either the supplier's name or the supplier number. If the supplier has never been used before, the option is given to place the 'new' supplier in the database. Since entering a new supplier is also an option on the main menu, it will be discussed later. For purposes of the example at hand, it is assumed that the desired supplier is already on file, and the user chooses to specify it by name, as in Screen 6.5. Since this process is nearly identical procedurally to selecting an account, a briefer example will be shown. The following screens will demonstrate the selection by 'name' of 'IBM' as the desired vendor.

Supplier Name: IBM
(Blank aborts process)

Screen 6.6 Looking for IBM

SELECTED SUPPLIER

IBM GOVT PRODUCTS
321 CYPRESS LANE
OAKMONT NM 43563
(505) 864-2169 Supplier #: 2170

Select an action: 1

- ```

- 1. Use this supplier -
- 2. Scroll to previous supplier -
- 3. Scroll to next supplier -
- 4. Return to last menu -
- 5. Return to main menu -

```

Screen 6.7 Selecting IBM

The IBM supplier displayed in Screen 6.7 is selected for use by choosing option '1' as shown. The supplier number shown is simply a number used by the database to keep track of the suppliers.

### C. REQUIRED DELIVERY DATE

Every type of order, except \$\$ Reimbursement to the RSS and transportation of material, presents the user with the opportunity to specify a Required Delivery Date (RDD). As shown in Screen 6.8, specification of the RDD is optional. While there are certainly no guarantees, an RDD will assist the Supply Department personnel in serving your needs.

#### REQUIRED DELIVERY DATE (RDD)

This is an optional 3 digit Julian date when the material or service is required. It is used to assist in the placement of the order.

Desire to specify an RDD (Y/N)? Y

Enter RDD: 233

Screen 6.8 Required Delivery Date



#### D. ORDER PRIORITY

Three different order priorities are offered users, as shown in Screen 6.9, and a brief explanation of each priority is presented.

##### ORDER PRIORITY

- A - Requirement is immediate and without the material required, the activity is unable to perform one or more of its primary missions.  
(Note: Requires NPS Superintendent approval)
- B - Requirement is immediate or it is known that such a requirement will occur in the immediate future.
- C - Routine requirement.

Select appropriate order priority:   C  

#### Screen 6.9 Order Priority

Most NPS requisitions are properly classified as priority 'C'. Occasionally sufficient justification is present to rate a requisition priority 'B'. Note that if priority 'A' is selected, Screen 6.10 will be displayed. If the user chooses priority 'A' and indicates that approval of the NPS Superintendent has not been obtained, the opportunity will be given to reselect a priority. For more information, phone the Issue/Receipt Control Branch at ext. 2012.

##### NOTICE

A priority code of 'A' requires  
NPS Superintendent approval.

Have you obtained approval (Y/N)?   

#### Screen 6.10 Priority 'A' Warning

## E. PREPARE PRINTER

Screen 6.11 will appear as a warning anytime output is being directed to the printer. If the user opts not to continue, the current procedure will be aborted, and the Main Menu, Screen 5.1, will be displayed. If a requisition was being prepared, all data entered for that requisition will be lost.

### PREPARE PRINTER

-----  
Since output will now be sent to the printer, it should be connected and have the power turned on now.

WARNING - Failure to prepare the printer may cause the computer to hang and possibly result in loss of data.

Continue (Y/N)? \_\_\_\_

#### Screen 6.11 Prepare Printer Warning

If the user continues past this screen, and the printer is not powered, the system may hang. This will result in the loss of the most recently entered data and may actually damage data files, rendering them unreadable.

If Screen 6.12 is displayed, ensure the printer is properly powered, connected, and loaded. The correct response to Screen 6.12 is an 'R'. If the printer is now prepared to print, the process will continue.

WARNING - If the user responds to Screen 6.12 with an 'A', the SRdb program will be aborted, the user returned to DOS, and data files may be damaged.

No paper error writing device PRN  
Abort, Retry, Ignore? \_\_\_\_

#### Screen 6.12 Device Error

Screens 6.13 and 6.14 are self-explanatory examples of displays which prompt loading of the correct form. The printer should be carefully loaded such that the document is positioned to print on the very first line. Forms loaded prior to the proper screen prompt may be ejected as the printer is addressed by the program.

LOAD SF-36 INTO PRINTER  
&  
PRESS ANY KEY TO CONTINUE

Screen 6.13    Load SF-36

LOAD DD-1348 INTO PRINTER  
&  
PRESS ANY KEY TO CONTINUE

Screen 6.14    Load DD-1348

## VII. TO PLACE AN ORDER

### TO PLACE AN ORDER

The purpose of this program is to generate the appropriate supply forms (DD 1348-6 &/or SF 36) to place an order.

Select the type of order

1. Open Purchase
2. Standard stock items from NPS RSS, NSC Oakland,  
or GSA
3. Standard forms from NSC Oakland
4. Publications from NAVPUBFORMCEN Phil
5. \$\$ Reimbursement to NPS Ready Supply Store
6. Transportation of material (e.g. Federal Express)
7. Return to main menu

Your order: \_\_\_\_

### Screen 7.1 To Place an Order

Screen 7.1 shows the 6 different categories of requisitions which may be prepared. The category titles should be self-explanatory. Common procedures, which were described in detail in the last chapter, will not be discussed further. These procedures, such as selecting an account or supplier, will be listed as occurring, however, the reader is referred to the previous chapter for details.

## A. OPEN PURCHASE

The Open Purchase order appears as option '1' on Screen 7.1. The following is the proper sequence of events to complete an open order.

- 1) Select an account
- 2) Select a supplier
- 3) Option to specify an RDD
- 4) Select requisition priority
- 5) Enter item to be ordered
- 6) Prepare printer

### 1. Enter Item to be Ordered

Screen 7.2 provides the user the opportunity to enter the description of the item to be purchased. It is important to note that although the entire description will appear on the requisition documents, only the 44 characters on the first line of the description will be saved in the database. It is important that the first line be sufficiently descriptive of the item, for it is this description which will appear on later reports and requisition summaries. SRdb has limited the size of the filed description due to program efficiency and disk space considerations. If the user leaves the first line of the item description blank, SRdb assumes that all desired items have been ordered. Up to 12 items may be ordered on a single SF-36. Note that this is a limitation of SRdb rather than an official supply limitation.

Item #: 1

ENTER ITEM TO BE ORDERED  
-----

Item description:

----- <--Only this line is filed

----- (Blank ends process.)

Unit of issue: ea

Unit price: \$ \_\_\_\_.

Quantity: \_\_\_\_

Is this order correct (Y/N)? \_\_\_\_

Screen 7.2 Open Order



Screen 7.3 is an example of an order for two IBM Personal Computers. If the user indicates that there is an error on the screen, an opportunity will be provided to edit the item currently displayed. Once the user indicates that the order for an item is correct, there is no opportunity to change the request. The entire requisition may be aborted just prior to printing, as described in Chapter 5.

Item #: 1

ENTER ITEM TO BE ORDERED  
-----

Item description:

IBM Personal Computer <--Only this line is filed

256K Memory

2 disk drives (Blank ends process.)

Unit of issue: ea

Unit price: \$ 1545.00

Quantity: 2

Is this order correct (Y/N)? Y

Screen 7.3 Sample Open Order

## B. STANDARD STOCK REQUISITION

The Standard Stock Order appears as option '2' on Screen 7.1 This option is used to place an order for any materials which have a National Stock Number (NSN) assigned, except forms or publications. The following is the proper sequence of events to complete a standard stock order.

- 1) Select an account
- 2) Select a stock source
- 3) Option to specify an RDD
- 4) Select requisition priority
- 5) Enter item to be ordered
- 6) Prepare printer

### 1. Select Standard Stock Source

As seen in Screen 7.4, there are 4 different sources to which a standard stock request may be addressed. The source used will depend upon where the item is located.

#### STANDARD STOCK REQUISITION

This program prepares a requisition for either a GSA or DOD standard stock item. If you want to order forms or publications, return to the previous menu.

- ```
-----  
- 1. NPS RSS Order -  
- 2. NSC Oakland Order -  
- 3. GSA Order -  
- 4. Return to previous menu -  
-----
```

Select: ____

Screen 7.4 Standard Stock Sources

2. Enter Item to be Ordered

Screen 7.5 provides the user the opportunity to enter the stock number and a short description of the item to be purchased. The item will be described in the SRdb database as a concatenation of the stock number and the short name. The Distribution COG may be obtained in the same listing in which the stock number was found.

Since a standard stock requisition may contain only one type of item, once the user verifies that the item description is correct, the requisition will be printed.

ENTER ITEM TO BE ORDERED

Stock number: 2342356664334

Short name: Pencil Holder
Black

Distribution COG symbol: 3I

Unit of issue: ea

Unit price: \$ 1.55

Quantity: 30

Is this order correct (Y/N)? Y

Screen 7.5 Standard Stock Item Entry

C. NSC OAKLAND FORMS REQUISITION

The requisition of forms from NSC Oakland, CA, appears as option '3' on Screen 7.1. This option is used to place an order for standard forms, which are assigned an NSN and are stocked at NSC Oakland. The following is the proper sequence of events to complete an NSC Oakland forms order.

- 1) Select an account
- 2) Option to specify an RDD
- 3) Select requisition priority
- 4) Enter item to be ordered
- 5) Prepare printer

Since this procedure is identical to the standard stock requisition process presented above, it will not be repeated. Screen 7.6 notifies the user of the selected process and provides an opportunity to abort.

FORMS ORDER

This program prepares the documents required to order forms from NSC OAKLAND. If you desire to order forms from NAVPUBFORMCEN Philadelphia, do not continue.

Continue (Y/N)?

Screen 7.6 Forms Order

D. PUBLICATION REQUISITION

The requisition of publications from NAVPUBFORMCEN, Philadelphia, PA, appears as option '4' on Screen 7.1. This option is used to place an order for publications, which are assigned an NSN and are stocked at NAVPUBFORMCEN. The following is the proper sequence of events to complete a publication order.

- 1) Select an account
- 2) Option to specify an RDD
- 3) Select requisition priority
- 4) Enter item to be ordered
- 5) Prepare printer

Since this procedure is identical to the standard stock requisition process presented above, it will not be repeated. Screen 7.7 notifies the user of the selected process and provides an opportunity to abort.

PUBLICATION ORDER

This program prepares the documents required to order publications from NAVPUBFORMCEN, Philadelphia.

Continue (Y/N)?__

Screen 7.7 Publication Order

E. RSS REIMBURSEMENT

The reimbursement of the NPS RSS through issuance of a 'money value only' DD-1348, appears as option '5' on Screen 7.1. The following is the proper sequence of events to complete a DD-1348 for RSS reimbursement:

- 1) Select an account
- 2) Enter the amount to be paid
- 3) Prepare printer

Screen 7.8 notifies the user of the selected process and provides an opportunity to abort.

\$\$ REIMBURSEMENT TO RSS

This program prepares a DD-1348 for 'money value only' as reimbursement to the NPS RSS.

Continue (Y/N)?__

Screen 7.8 RSS Reimbursement

F. TRANSPORTATION OF MATERIAL REQUISITION

The requisition of transportation of material appears as option '6' on Screen 7.1. This option is used to to pay for services, such as Federal Express, which are used to transport needed materials. The following is the proper sequence of events to complete a transportation of materials requisition.

- 1) Select an account
- 2) Select a supplier (company)
- 3) Enter transportation serial number
- 4) Select requisition priority
- 5) Enter to whom item is addressed
- 6) Prepare printer

TRANSPORTATION OF MATERIAL

This program prepares the supply documents necessary to use commercial carriers (e.g. Federal Express or UPS) to ship items.

Continue (Y/N)?__

Screen 7.9 Transportation of Material

Screen 7.9 notifies the user of the selected process and provides an opportunity to abort. The only new procedure in this order sequence is the entering of the Transportation Document Serial Number as displayed in Screens 7.10 and 7.11. Screen 7.10 provides the opportunity to abort the process, if the proper document serial number has not been obtained.

ENTER TRANSPORTATION SERIAL NUMBER

Transportation requests require issuance of a special
serial number directly from the comptroller. Phone
the NPS comptroller at x2257 to obtain.

Select an action: 1

- - 1. Ready to enter serial number -
- 2. Abort to main menu -

Screen 7.10 Prepare to Enter Transportation Number

Enter Serial #: 1234

Correct (Y/N)?

Screen 7.11 Enter Transportation Serial Number

VIII. UPDATE FILES WITH PURCHASE ORDER

In choosing option 2 from the Main Menu, Screen 5.1, the user is presented with the opportunity to enter purchase order numbers from supply for each item. Screen 8.1 notifies the user of the selected process and provides an opportunity to abort.

INPUT PURCHASE ORDER NUMBER (PO#) -----

Once a PO# is received from supply, this program permits entry of the PO# for each item ordered. Since not all items on the original requisition may have been ordered by supply on the same purchase order, you are requested to separately confirm each item on the purchase order.

Continue (Y/N)? ____

Screen 8.1 Input P.O. #

If the user chooses to continue, he will be asked to input the requisition number and the purchase order number (PO #) of the item he is seeking, as in Screen 8.2. Leaving the requisition number blank will abort the process and return the user to the main menu.

ENTER DATA -----

Requisition # : 23423423
(Blank ends process)

Purchase Order #: 12345678

Correct (Y/N)? ____

Screen 8.2 Input Req # and PO #

If the requisition is not found in the file, Screen 8.3 is displayed. The user will subsequently be given another opportunity to enter the requisition number.

REQUISITION NOT IN FILE

Please double check the requisition # and try again.

Press any key to continue...

Screen 8.3 Requisition Not Found

Once the requisition is found, each item on the requisition will be displayed one at a time. The user will have the opportunity to indicate exactly which items are on the purchase order, as seen in Screen 8.4.

Requisition #: 23423423

Item: dBASE III

Quantity: 1

Unit Price: 456.75

Is this item on purchase # 12345678 (Y/N)?__

Screen 8.4 Item Displayed for PO # Input

IX. MARK ITEMS RECEIVED

In choosing option 3 from the Main Menu, Screen 5.1, the user is presented with the opportunity to enter notation of the receipt of items in the database. Screen 9.1 notifies the user of the selected process and provides an opportunity to abort.

INPUT RECEIPT STATUS

You will be given the opportunity to indicate if specific items, currently not marked as received, have now been received. Items reviewed will be grouped by requisition number or purchase order number at your option.

Continue (Y/N)? ____

Screen 9.1 Mark Receipt of an Item

The user is permitted to identify the received item by specifying the original requisition number or the purchase order number (PO#), assuming the file has been updated with the PO#. Screen 9.2 shows an example of the user choosing to specify a requisition number.

SRdb will then screen the database and individually display each item filed under that number which is not already marked as received. If all items have been marked as received, the user will be so informed. Otherwise, each outstanding item will be displayed, as in Screen 9.3.

PREPARING TO MARK ITEMS RECEIVED

Select: 1

- 1. Show by requisition # -
- 2. Show by purchase order # -
- 3. Return to main menu -

Enter Requisition #: 53156504

Note - Blank ends process

Screen 9.2 Pick Requisition Number or PO #

The following item shown as not received

Requisition #: 53156504

Purchase Ord #: 54443221

Item: dBASE III

Quantity: 1 Unit Price: \$ 456.75

Has this item been received (Y/N)? ____

Screen 9.3 Item Not Shown as Received

X. LIST ITEMS NOT YET RECEIVED

In choosing option 4 from the Main Menu, Screen 5.1, the user is presented with the opportunity to create a listing of all items in the database which are tagged as not received. Screen 10.1 notifies the user of the selected process and provides an opportunity to abort.

LIST ITEMS NOT RECEIVED

This program generates a listing of items ordered but not yet received. The list may be sent to the screen or printer at the user's option. The search for items not received may also be limited within parameters provided by the user.

Continue (Y/N)?

Screen 10.1 List Items Not Received

As shown in Screen 10.2, the user may select from two search types. Both searches will list only items marked as not received. The user is also given the option of displaying the listing on the screen or routing it to the printer.

Select Search Type: ____

- 1. No limits (show all items not received) -
- 2. Limit list to specific account number -
- 3. Return to main menu -

Select Display: ____

- 1. Screen -
- 2. Printer -

Screen 10.2 Search Type Selection

XI. DISPLAY A REQUISITION

In choosing option 5 from the Main Menu, Screen 5.1, the user is presented with the opportunity to display one or more requisitions from the SRdb database. Screen 11.1 notifies the user of the selected process and provides an opportunity to abort.

DISPLAY REQUISITION

This program will display one or more requisitions subject to user selection criteria. The list may be sent to the screen or printer at the user's option.

Continue (Y/N)? ____

Screen 11.1 Display Requisitions

Screen 11.2 is the primary menu for selecting options for the display of requisitions. The user is asked to choose the type of requisition search and the mode of the output, either screen or printer.

If the user wants to print out only a single requisition, he will select option '1' from Screen 11.2, and he will then be prompted for entry of the requisition number. Option '2' from Screen 11.2 will allow the user to view all requisitions on file from a specific account. If the user selects option '4', he will have the opportunity to review all requisitions from a single account generated between two dates, as seen in Screen 11.3.

DISPLAY REQUISITIONS

Selection: 3

```
=====
0.  Return to main menu
1.  Display single requisition
2.  Display requisitions from specific account
3.  Display requisitions subject to account # & date
=====
```

Select Display: 1

```
=====
1.  Screen
2.  Printer
=====
```

Screen 11.2 Display Requisition Options

Find Requisitions Created

Between

10/01/85 and 12/11/85
mm/dd/yy mm/dd/yy

Screen 11.3 Requisition Generation Dates

XII. CHANGE AN ITEM PRICE

In choosing option 6 from the Main Menu, Screen 5.1, the user is presented with the opportunity to update the actual price of an item which has already been ordered and is in the SRdb database. It is not uncommon to receive price changes from NPS Supply after an item has been ordered. Screen 12.1 notifies the user of the selected process and provides an opportunity to abort.

CHANGE ITEM PRICE -----

This module is used when the item's final price differs from that on the original requisition. You will be shown all items associated with a specific requisition # or purchase order #. Upon displaying each item, you will be given the opportunity to change the item's unit price or scroll to the next item.

Continue (Y/N)? ____

Screen 12.1 Change Item Price

Items may be reviewed by specifying either the original requisition number or the purchase order number, if one has been assigned. An example will be shown specifying a requisition number, as in Screen 12.2.

If the requisition number entered in Screen 12.3 cannot be found, Screen 12.4 will be presented. The user will be presented with another opportunity to enter the correct requisition number.

Screen 12.5 presents the item to the user and asks if a price change is desired. If the user responds 'Y', then the cursor will move to the price and permit a change. The user will be asked to confirm changes before they are saved to the database. If a mistake is made, the process may be repeated until the correct price is filed.

Preparing to change an item price

Select: 1

```
-----  
- 1. Show by requisition # -  
- 2. Show by purchase order # -  
- 3. Return to main menu -  
-----
```

Screen 12.2 Select Item by Req # or PO #

Enter Requisition # : 12345678

Note - Blank ends process

Correct (Y/N)?

Screen 12.3 Enter Requisition Number

Requisition # not in file.
Please double check the Requisition # and try again.
Press any key to continue...

Screen 12.4 Requisition Not Found

Requisition #: 53365707

Purchase Ord #:

Item: 9 Meg Buffer Board

Quantity: 2 Unit Price: \$ 2875.50

Change unit price (Y/N)? ____

Screen 12.5 Item Displayed for Price Change

XIII. ENTER A NEW SUPPLIER

In choosing option 7 from the Main Menu, Screen 5.1, the user is presented with the opportunity to enter a new supplier into the database. All required information, name, address, and phone number should be available to the user. The phone number is mandatory. It is used by SRdb to select a supplier number. The supplier number is simply a number used by the program to keep track of the suppliers. Since suppliers may be recalled by their name, there is no need for a user to commit the supplier numbers to memory or maintain an external list. Screen 13.1 notifies the user of the selected process and provides an opportunity to abort.

INPUT A NEW SUPPLIER -----

In order to input a new supplier, you must have the following information available at this time:

Supplier Name
Address
Phone #

The phone number is mandatory. Do not proceed unless you are ready to enter the above information.

Continue (Y/N) ? ____

Screen 13.1 Enter New Supplier

Screen 13.2 shows an example of the blanks the the user will be expected to complete. Screen 13.3 is an example which has been completed by the user. Note that SRdb has assigned the supplier number.

INPUT NEW SUPPLIER

Name: _____

Address: _____

City: _____ State: _____ Zip: _____

Phone: (____) ____-____

Note - Phone # is mandatory

Screen 13.2 Input new Supplier

INPUT NEW SUPPLIER

Name: New Guy Cables

Address: 1021 Brandnew Circle

City: Jigdog State: GA Zip: 31907

Phone: (404) 421-8895

Assigned Supplier #: 8896

Press any key to continue...

Screen 13.3 New Supplier Example

XIV. CREATE A NEW ACCOUNT

In choosing option 8 from the Main Menu, Screen 5.1, the user is presented with the opportunity to enter a new account into the database. All required information should have been received from the NPS Comptroller office and be available to the user. Screen 14.1 notifies the user of the selected process and provides an opportunity to abort.

CREATE A NEW ACCOUNT -----

In order to enter a new account you must have the following information available at this time:

Account Name
Account # (assigned by NPS Comptroller)
Serial # range assigned to account
Is it a research account?

WARNING - If you do not understand any of the above items, seek assistance and do not continue further at this time.

Continue (Y/N)? ____

Screen 14.1 Enter New Account

Screen 14.2 shows an example of the blanks the the user will be expected to complete. Screen 14.3 is an example which has been completed by the user. The ability to enter the last serial number used assists in transitioning current accounts to the SRdb system. If an account is brand new, the next document serial number to be used is the first one in the series. If the account is being transitioned to SRdb, simply enter the last serial number used, and the system will properly number future requisition documents.

INPUT NEW ACCOUNT

Account Name: _____

Account Number: _____

Code: _____

Research Account (T/F): ____

Requisition Serial # Range - From: _____
To: _____

Last Serial # Used: _____
(leave blank if account never used yet)

Screen 14.2 Input New Account Screen

INPUT NEW ACCOUNT

Account Name: Newman, B N

Account Number: R4321

Code: 54NM

Research Account (T/F): T

Requisition Serial # Range - From: 9000
To: 9005

Last Serial # Used: _____
(leave blank if account never used yet)

Correct (Y/N)? Y

Screen 14.3 New Account Example

APPENDIX C DATABASE STRUCTURE

The program uses 5 separate database files (.dbf), 4 of which have 2 indexes each. The width of fields and indexes are expressed in bytes. It should be noted that the size of indexes are approximations which were calculated in accordance with the procedures outlined in the Ashton Tate - Advanced Programmer's Guide [Ref. 12]. The grand totals given for each account represent the minimum number of bytes required for each record in the database. The reader is referred to the Ashton Tate - Advanced Programmer's Guide [Ref. 12] for more in depth sizing algorithms.

ACCNT.dbf

<u>Field</u>	<u>Type</u>	<u>Width</u>
ANAME	C	12
ACCNO	C	5
CODE	C	4
RESEARCH	L	1
LOSERNO	C	4
HISERNO	C	4
LASTUSED	C	4
Total		<u>34</u>
<u>Indexes</u>		
ANAME_A	20	
ACCNO_A	13	<u>33</u>
Grand total		<u>67</u>

SUP.dbf

<u>Field</u>	<u>Type</u>	<u>Width</u>
SNAME	C	20
ADDR	C	20
CITY	C	12
STATE	C	2
ZIP	C	5
FONE	C	10
SUPNO	C	4
Total		<u>75</u>
<u>Indexes</u>		
SNAME_S	28	
SUPNO_S	12	<u>40</u>
Grand total		<u>115</u>

REQ. dbf

<u>Field</u>	<u>Type</u>	<u>Width</u>
REONO	C	8
ACCNO	C	5
PRI CODE	C	1
SUPNO	C	4
RDATE	D	9
Total		<u>27</u>
<u>Indexes</u>		
REONO_R	16	
ACCNO_R	13	
Grand total		<u>29</u>

ITEM. dbf

<u>Field</u>	<u>Type</u>	<u>Width</u>	<u>Dec</u>
DESCRIP	C	44	
UNIT	C	2	
QUANT	N	5	
UPRICE	N	8	2
REONO	C	8	
PONO	C	8	
RCVD	L	1	
Total		<u>76</u>	
<u>Indexes</u>			
REONO_I	16		
SUPNO_I	16		
Grand total		<u>32</u>	
		<u>108</u>	

TEMPLATE. dbf

<u>Field</u>	<u>Type</u>	<u>Width</u>	<u>Dec</u>
ITEMNO	N	2	
DESCRIP	C	44	
DESCRIP2	C	44	
DESCRIP3	C	44	
UNIT	N	2	
UPRICE	N	8	2
QUANT	N	8	
Total		<u>155</u>	

No indexes are required. This .dbf serves as a template which is copied to a TEMPORD.dbf whenever an open purchase order is to be made. After the requisition documents have been prepared, TEMPORD.dbf is erased.

APPENDIX D

DATA DICTIONARY

I. INTRODUCTION

Memory variables in dBASE III remain local to the modules in which they were created unless they were declared "PUBLIC" prior to having a value initially assigned. This data dictionary lists all PUBLIC memory variables. Unless otherwise noted, memory variables are assumed to be of type character.

II. STATIC DATA

Static data elements usually do not change values during a typical normal run. They are frequently associated with control or reference. All static data elements have preassigned values.

A. File Names

The following data elements identify specific disk files and are always concatenated with a disk identifier (A,B,C, etc. stored as C_ADRV or C_SDRV).

ACCNO_A	ndx file to ACCNT.dbf indexed on account number.
ACCNO_R	ndx file to REQ.dbf indexed on account number.
ACCNT	dbf file for all accounts.
C_ADRV	Memory variable stored in CONFIG.VAR file designating the expected disk drive location for account and requisition related files.
C_SDRV	Memory variable stored in CONFIG.VAR file designating the expected disk drive location for supplier files.
ITEM	dbf file holding each item which has been requisitioned and related information.
PONO_I	ndx file to ITEM.dbf indexed on purchase order numbers.
REQ	dbf file holding data relating requisitions to specific suppliers and accounts.
REQNO_I	ndx file to ITEM.dbf indexed on requisition numbers.
REQNO_R	ndx file to REQ.dbf indexed on requisition numbers.
SNAME_S	ndx file to SUP.dbf indexed on supplier name.
SUP	dbf file listing names, addresses, and phone numbers of known suppliers.

SUPNO_S ndx file to SUP.dbf indexed on supplier number.

TEMPLATE an empty dbf file containing the structure of fields essential in creation of an open order requisition. It is used to create the TEMPORD file whenever an open order requisition is executed.

TEMPORD a temporary file used to hold the full description of all items in an open order. The regular ITEM.dbf only stores a one line description (38 characters). This file temporarily holds 2 additional 38 character lines until the SF-36 has been printed. After the SF-36 document has been successfully printed, and the essential details of the requisition have been filed in the ITEM.dbf and REQ.dbf, the TEMPORD file is erased.

B. Misc Static Memory Variables

CCOLOR Holds color assignments for normal text screen output. May be changed to accommodate monochrome monitors based on the value of C_CLRMON, however, since the program is expected to always be run on the same system, this is viewed as a static variable.

ERRCOLOR Holds color assignments for error messages for screen output. May be changed to accommodate monochrome monitors based on the value of C_CLRMON, however, since the program is expected to always be run on the same system, this is viewed as a static variable.

FY Last digit of the fiscal year used in the construction of the appropriate FY oriented julian date for use in forming valid requisition numbers. The fiscal year is stored in a FY.ID file on the account disk.

MONEY String "MONEY VALUE ONLY" to be inserted on on a DD-1348 prepared to pay an NPS RSS billing.

MSCOLOR Color assignments for highlighted messages for screen output. May be changed to accommodate monochrome monitors based on the value of C_CLRMON, however, since the program is expected to always be run on the same system, this is viewed as a static variable.

WMSG Wait message which requests user to "press a key to continue. This was used vice the system wait prompt because this message is centered.

III. DYNAMIC INPUT DATA ELEMENTS

The following data elements directly receive values from user inputs during the program run. A brief description of the variable is followed by legal value constraints (length, data type).

ACCNO	Account number assigned by NPS Comptroller and entered in ACCNT.dbf in NEW_ACNT.PRG; is unique. (5 char)
ADDR	Supplier street address as input to SUP.dbf by user; filed in SUP.dbf by NEW_SUPL.PRG (20 char)
ADDR1	1st line of address of receiver of material shipped by XPOR_ORD.PRG. (25 char)
ADDR2	2nd line of address of receiver of material shipped by XPOR_ORD.PRG. (25 char)
ADDR3	3rd line of address of receiver of material shipped by XPOR_ORD.PRG. (25 char)
ADDR2	4th line of address of receiver of material shipped by XPOR_ORD.PRG. (25 char)
ANAME	Account name, usually individual's last name, as entered in ACCNT.dbf in NEW_ACNT.PRG. (12 char)
AREACODE	Phone areacode of supplier as input to SUP.dbf in NEW_SUPL.PRG. (3 char)
CITY	City of supplier address as input to SUP.dbf by user; filed in SUP.dbf by NEW_SUPL.PRG (12 char)
CODE	Code of individual account as entered in ACCNT.dbf in NEW_ACNT.PRG. (4 char)
DESCRP	The mem var which receives an item description. The value is subsequently filed in the DESCRIP field of the ITEM.dbf file. (44 char)
DIST	Distribution symbol used in ordering stock-numbered items. (2 char)
EXT1	Phone extention of the 1st point of contact listed in a supplemental memo accompanying an order exceeding \$1,000. (4 char)
EXT2	Phone extention of the 2nd point of contact listed in a supplemental memo accompanying an order exceeding \$1,000. (4 char)
FONE	Phone number of supplier as input to SUP.dbf in NEW_SUPL.PRG. (7 char)
HISERNO	High serial number in range of serial numbers assigned to a specific account by the NPS Comptroller and entered in ACCNT.dbf in NEW_ACNT.PRG. (4 char)

LASTUSED Serial number used in most recent requisition (not including transportation requisitions) from a specific account entered in ACCNT.dbf in NEW_ACNT.PRG. This field is updated in the process of making each new order. (4 char)

LOSERNO Low serial number in range of serial numbers assigned to a specific account by the NPS Comptroller and entered in ACCNT.dbf in NEW_ACNT.PRG. (4 char)

NOMEN 1 Used to hold 1st line of a short description for a stock-numbered items in BUY1_ITM.PRG. (13 char)

NOMEN 2 Used to hold 2nd line of a short description for a stock-numbered item in BUY1_ITM.PRG. (13 char)

POC1 1st "point of contact" listed in a supplemental memo which accompanies orders exceeding \$1,000. (22 char)

POC2 2nd "point of contact" listed in a supplemental memo which accompanies orders exceeding \$1,000. (22 char)

PRI 1 letter code indicating priority assigned to requisition by user. ('A', 'B', or 'C')

QUNT Quantity of item to be ordered. (1 - 99999, numeric)

RESEARCH Logical variable assigned .T. if account involves research monies. (1 logical)

RDD Required delivery date; 3 digit julian date that material is desired; assists supply department in placing order. (1 - 366, numeric)

RDD_YES Logical variable showing if the user has chosen to specify an RDD. (.T. if rdd specified; .F. if rdd not specified)

SHIPTO Addressee to whom material is to be transported when requisition is to cover transportation of material; used by XPOR_ORD.PRG (25 char)

STATE State of supplier address as input to SUP.dbf by user; filed in SUP.dbf by NEW_SUPL.PRG (2 char)

STKNUM Federal stock number of item ordered by BUY1_ITM.PRG (13 char)

TODAY Date variable holding today's date. If transactions are to be generated for dates other than today, this date must be modified in GET_DATE.PRG (MM/DD/YY, date)

UNT Unit of order, e.g. 'ea', 'dz', etc. (2 char)

UPRCE Unit price of item. Must allow for order of items
 which are free. (0 - 99,999.99, numeric)

ZIP Zip code of supplier address as input to SUP.dbf
 by user; filed in SUP.dbf by NEW_SUPL.PRG
 (5 char)

IV. DYNAMIC OUTPUT DATA ELEMENTS

The following data elements receive values from within the program during the program run. The domain of values is controlled since the user never inputs a value directly, but rather, values are assigned internally based upon his response. An example of this type of data element may be seen in the following construction in which DEMAND is a dynamic output data element:

```
      If answer = 'Y'
        DEMAND = 'R'
      else
        DEMAND = 'N'
      endif
```

AREACODE1 Areacode of 2nd supplier to appear on a supplemental memo which accompanies orders exceeding \$1,000.00. AREACODE1 is not input by user, but rather read from AREACODE field of the record the user has chosen from the SUP.dbf. Value assignment is made in the SUP_MEMO.PRG (3 char)

AREACODE2 Areacode of 3rd supplier to appear on a supplemental memo which accompanies orders exceeding \$1,000.00. AREACODE2 is not input by user, but rather read from AREACODE field of the record the user has chosen from the SUP.dbf. Value assignment is made in the SUP_MEMO.PRG (3 char)

CHECKER an internal variable which assumes the value of the LASTUSED serial number of an account, used in the calculation of a new requisition number by CALC_REQ.PRG (4 char)

CITY1 City of 2nd supplier to appear on a supplemental memo which accompanies orders exceeding \$1,000.00. CITY1 is not input by user, but rather read from CITY field of the record the user has chosen from the SUP.dbf. Value assignment is made in the SUP_MEMO.PRG (12 char)

CITY2 City of 3rd supplier to appear on a supplemental memo which accompanies orders exceeding \$1,000.00. CITY2 is not input by user, but rather read from CITY field of the record the user has chosen from the SUP.dbf. Value assignment is made in the SUP_MEMO.PRG (12 char)

COUNTER Numeric variable used to count the number of line items in a requisition. (1 - 13, numeric)

DEMAND Refers to the recurring nature of demand for an item or service. ('R' if demand is recurring; 'N' if nonrecurring.)

DOCID Document identifier as per p.4-7 of Ref 1. ('AOA' for NSN items; 'AOD' for Navy Forms & Publications; 'AOE' for non-standard stock items.)

FONE1 Phone number of 2nd supplier to appear on a supplemental memo which accompanies orders exceeding \$1,000.00. FONE1 is not input by user, but rather read from FONE field of the record the user has chosen from the SUP.dbf. Value assignment is made in the SUP_MEMO.PRG (7 char)

FONE2 Phone number of 3rd supplier to appear on a supplemental memo which accompanies orders exceeding \$1,000.00. FONE2 is not input by user, but rather read from FONE field of the record the user has chosen from the SUP.dbf. Value assignment is made in the SUP_MEMO.PRG (7 char)

FUND Fund Code as described on p. 4-8 of Ref. 1. ('2S' for non-reimbursible requisitions; 'Y6' for APA material)

GOAHEAD Logical control value used to control execution flow in BUY2_ITM.PRG after running of WARNING.PRG (1, logical)

JULIAN 3 digit numeric value based on julian date as calculated by GET_DATE.PRG (1 - 366, numeric)

MARKER Tracks the number of lines being displayed on the screen to ensure that output does not overflow screen. Initialized to 9, triggers a 'wait' when value > 23. (9-24, numeric)

MS_Code M&S Code as described on p. 4-7 of Ref. 1. Value is assigned based on PRI_CODE. Used only for standard stock requisitions. ('W' for PRI_CODE = 'A'; 'T' for PRI_CODE = 'B' or 'C').

PRIORITY Priority code as described on p. 4-8 of Ref. 1. Value is assigned based on PRI_CODE. ('08' for PRI_CODE = 'A'; '10' for PRI_CODE = 'B'; '15' for PRI_CODE = 'C').

REQNO Requisition number which is concatenation of the last digit of the FY + 3-digit JULIAN date + 4-digit serial number which is taken from an individual's account. (8 char)

RMKO Remarks which are printed in block 'O' of a DD-1348. Consists of expense element appropriate to the type of material ordered as per Ref. 6. (Valid assignments are 'T', 'W', 'P', 'R', 'Q', and 'L'.)

ROUTE Routing identifier as described on p. 4-7 of Ref. 1. Used only for standard stock requisitions. ('NOZ' for NSC Oakland, 'NFZ' for NAVPUBFORMCEN, Phil, 'Z9S' for NPS RSS).

SENDTO Supplier name from SNAME field of record selected by user from SUP.dbf. Used in Block A of DD-1348. (20 char)

SERVICE Last digit of current FY. Entered in block 45 of DD-1348. (0-9, char)

SF36 Logical variable = .T. when printing of an SF-36 is required. An open order requires printing of an SF-36 when
 1) Item description exceeds available space on DD-1348 (> 22 char).
 2) More than one item is being ordered on same open order requisition.
 (1, logical)

SIGNAL Signal Code as described on p. 4-8 of Ref. 1. ('D' for free publications, 'A' for all other items).

SNAME1 Name of the 2nd supplier to appear on a supplemental memo which accompanies orders exceeding \$1,000.00. SNAME1 is not input by user, but rather read from SNAME field of the record the user has chosen from the SUP.dbf. Value assignment is made in the SUP_MEMO.PRG (20 char)

SNAME2 Name of the 3rd supplier to appear on a supplemental memo which accompanies orders exceeding \$1,000.00. SNAME2 is not input by user, but rather read from SNAME field of the record the user has chosen from the SUP.dbf. Value assignment is made in the SUP_MEMO.PRG (20 char)

STATE1 State address of the 2nd supplier to appear on a supplemental memo which accompanies orders exceeding \$1,000.00. STATE1 is not input by user, but rather read from STATE field of the record the user has chosen from the SUP.dbf. Value assignment is made in the SUP_MEMO.PRG (2 char)

STATE2 State address of the 3rd supplier to appear on a supplemental memo which accompanies orders exceeding \$1,000.00. STATE2 is not input by user, but rather read from STATE field of the record the user has chosen from the SUP.dbf. Value assignment is made in the SUP_MEMO.PRG (2 char)

SUPADR Supplementary address of requisition as described on p. 4-8 of Ref. 1. This is also known as the job order number. It results from a rather series of concatenations performed in GET_CAT.PRG (5 char)

SUPNO Unique supplier number on which the SUP.dbf is indexed. It is assigned at the time a new supplier is entered in the SUP.dbf by NEW_SUPL.PRG. It is based upon an incrementing of the last 4-digits of the supplier's phone number. The incrementation continues until an unused number is found. (4 char)

TOTVAL Total value of a specific requisition or series of items. Value range depends upon specific usage, however, in general (.01 - 99,999.99, numeric).

TYPEORDR Type of order as assigned by NEW_ORDR.PRG ('OPEN', 'STOCK', 'FORM', 'PUB', 'RSS\$', or 'XPORT').

ZIP1 Zip code of the 2nd supplier to appear on a supplemental memo which accompanies orders exceeding \$1,000.00. ZIP1 is not input by user, but rather read from ZIP field of of the record the user has chosen from the SUP.dbf. Value assignment is made in the SUP_MEMO.PRG (5 char)

ZIP2 Zip code of the 3rd supplier to appear on a supplemental memo which accompanies orders exceeding \$1,000.00. ZIP2 is not input by user, but rather read from ZIP field of of the record the user has chosen from the SUP.dbf. Value assignment is made in the SUP_MEMO.PRG (5 char)

APPENDIX E
SRdb SOURCE CODE LISTINGS

The following is a listing of the 40 SRdb modules, which are written in dBASEIII application language. Each module is prefaced by a prologue giving the name, author, purpose, and a summary of interaction with other modules.

BUY1_ITM. PRG	NEW_ORDR. PRG
BUY2_ITM. PRG	NEW_SUPL. PRG
CALC_REQ. PRG	OPN_ORDR. PRG
CHG_PRICE. PRG	PRNT_SMO. PRG
CONFIG. PRG	PRTSF36. PRG
DISP_REQ. PRG	PRT1348a. PRG
DIS1_REQ. PRG	PRT1348o. PRG
DIS2_REQ. PRG	PRT1348x. PRG
DIS3_REQ. PRG	PUB_ORD. PRG
FILE_ITM. PRG	QUIT. PRG
FILE_REQ. PRG	REQ_HDR. PRG
FORM_ORD. PRG	RSS_ORD. PRG
GET_CAT. PRG	SEL_ACNT. PRG
GET_DATE. PRG	SEL_SUPL. PRG
GET_PRI. PRG	SHW_NRI. PRG
GET_RDD. PRG	STK_ORDR. PRG
IN_PONO. PRG	SUP_MEMO. PRG
IN_RCVD. PRG	WARNING. PRG
MAIN. PRG	WARN_PRT. PRG
NEW_ACNT. PRG	WELCOME. PRG


```

*****
*
* Module Name:  BUY1_ITM.PRG
*
* Author:  Tom Trotter
*
* Purpose:  This module generates a screen for user input of a single
*           item requisition.  It is invoked in situations when only
*           one item may be purchased per requisition # (e.g. when
*           making a DOD,GSA, or NPS RSS stock requisition).
*
* This module is called by:  FORM_ORD.PRG
*                           PUB_ORD.PRG
*                           STK.ORDR.PRG
*
* This module calls:  N / A
*
*****

```

```

ok = ' '
do while ok <> 'Y'
  public descrp,unt,uprce,qunt,dist,stknum,nomen1,nomen2
  clear
  @ 2,24 SAY "ENTER ITEM TO BE ORDERED"
  @ 3,24 SAY "-----"
  stknum = ' '
  nomen1 = ' '
  nomen2 = ' '
  dist = ' '
  @ 5,14 SAY "Stock number:"
  @ 5,29 GET stknum picture "@!"
  @ 7,14 SAY "Short name:"
  @ 7,27 GET nomen1
  @ 8,27 GET nomen2
  @ 10,14 SAY "Distribution COG symbol:"
  @ 10,40 GET dist picture "@!"
  read
  descrp = nomen1 + " " + stknum
  unt = 'ea'
  uprce = 0.00
  qunt = 0
  @ 12,14 SAY "Unit of issue:"
  @ 12,30 GET unt picture "@A"
  @ 14,14 SAY "Quantity:"
  @ 14,25 GET qunt picture "99999"
  if typeordr <> 'PUB'
    @ 16,14 SAY "Unit price: $"
    @ 16,28 GET uprce picture "99999.99"
  endif
  read
  set color to &mcolor
  @ 20,21 SAY "Is this order correct (Y/N)?"
  @ 20,51 GET ok picture "@!"
  set color to &ccolor
  read
enddo
return

```

```

*****
* Module Name:  BUY2_ITM.PRG                                *
* Author:  Tom Trotter                                       *
* Purpose:  This module is invoked to accept multiple items per single *
*            requisition as in the case of an open purchase. The user *
*            is limited to only 12 items per requisition due to the *
*            physical size of a SF-36 document. There is currently no *
*            provision for generation of multiple SF-36's per requisition. *
*            Each item is saved to the item.dbf after it is certified to *
*            be correct by the user. The process will stop when the *
*            user leaves the item description blank.          *
* This module is called by:  OPN_ORDR.PRG                    *
* This module calls:  FILE_ITM.PRG                           *
*****

```

```

public unt,uprce,totval,qunt,goahead
totval = 0.00
ok = ' '
do while counter < 13
  goahead = .T.
  clear
  @ 1,4 SAY "Item #:"
  @ 1,12 SAY counter picture "##"
  @ 2,24 SAY "ENTER ITEM TO BE ORDERED"
  @ 3,24 SAY "-----"
  descrp = '
  descrp2 = '
  descrp3 = '
  @ 5,1 SAY "Item description:"
  @ 6,1 GET descrp
  @ 6,41 SAY "<--Only this line is filed"
  @ 7,1 GET descrp2
  @ 8,1 GET descrp3
  @ 8,46 SAY "(Blank ends process.)"
  read
  if descrp = ' ' .and. counter = 1
    do warning
  endif
  if goahead
    if descrp = ' '
      return
    endif
    unt = 'ea'
    uprce = 0.00
    qunt = 0
    @ 11,14 SAY "Unit of issue:"
    @ 11,30 GET unt picture "A"
    @ 13,14 SAY "Unit price: $"
    @ 13,28 GET uprce picture "99999.99"
    @ 15,14 SAY "Quantity:"
    @ 15,25 GET qunt picture "99999"
    read
    set color to &mcolor
    @ 20,21 SAY "Is this order correct (Y/N)?"
    @ 20,51 GET ok picture "A!"
    set color to &ccolor
    read
    if ok = 'Y' .and. qunt <> 0

      totval = totval + (uprce * qunt)
      do file_itm
      if counter = 1
        select E

```

```

        use &template
        copy to &tempord
        use &tempord
    else
        select E
    endif
    append blank
    replace itemno with counter
    replace descrip with M->descrip
    replace descrip2 with M->descrip2
    replace descrip3 with M->descrip3
    replace unit with M->unt
    replace quant with M->qunt
    replace uprice with M->uprice
    counter = counter + 1
endif
if ok = 'Y' .and. qunt = 0
    @ 22,15 SAY "Sorry, but ordering '0' quantity makes no sense."
    set color to &mscolor
    wait &wmsg
    set color to &ccolor
endif
endif
enddo
return

```

```

*****
* Module Name:  CALC_REQ.PRG
*
* Author: Tom Trotter
*
* Purpose:  This module calculates and assigns the appropriate
*           requisition number (reqno) for each new requisition.
*           An 8-digit reqno is comprised of a 4-digit julian date,
*           and the last 4 digits will be the next serial number to
*           be used for a specific account. The only exception is
*           when the requisition is for transportation expenses, in
*           which case the user is prompted to phone the NPS
*           Comptroller to have the last 4 digits assigned.
*
* This module is called by:  NEW_ORDR.PRG
*
* This module calls:  N / A
*
*****

```

```

public reqno
reqno = str(year(today)-1980,1)
if year(today) > 1989
    reqno = str(year(today)-1990,1)
endif
reqno = reqno + substr(str(julian+1000,4),2)
if typeodr = 'XPORT'
    ok = ' '
    ans = 2
    do while ok <> 'Y'
        set scoreboard on
        clear
        @ 6,20 SAY "ENTER TRANSPORTATION SERIAL NUMBER"
        @ 7,20 SAY "-----"
        @ 9,16 SAY "Transportation requests require issuance of a special"
        @ 10,16 SAY "serial number directly from the comptroller. Phone"
        @ 11,16 SAY "the NPS comptroller at x2257 to obtain."
        @ 13,24 SAY "Select an action: "
        @ 14,24 SAY "-----"
        @ 15,20 SAY "1. Ready to enter serial number"
        @ 16,20 SAY "2. Abort to main menu"
        @ 13,43 GET ans picture "#" range 1,2
        read
        set scoreboard off
        do case
            case ans = 1
                xportno = ' '
                @ 13,0 clear
                @ 13,24 SAY "Enter Serial #:"
                @ 13,40 get xportno picture "XXXX"
                read
                set color to &mscolor
                @ 15,24 SAY "Correct (Y/N)?"
                set color to &ccolor
                @ 15,39 GET ok picture "@!"
                read
                reqno = reqno + xportno
            case ans = 2
                return to master
        endcase
    enddo
else
    public checker
    checker = A->lastused
    check = val(checker) + 1
    if check > val(A->hiserno)
        checker = A->loserno
    else

```

```
        checker = str(check,4)
    endif
    reqno = reqno + checker
endif
return
```



```

*****
*
* Module Name: CHG_PRICE.PRG
*
* Author: Tom Trotter
*
* Purpose: This module allows the user to change the price of an item
*          if a price revision is received. It is common for items to
*          be received at different prices than those used at the time
*          of ordering.
*
* This module is called by: MAIN.PRG
*
* This module calls: N / A
*
*****

```

```

ans = ' '
clear
@ 5,29 SAY "CHANGE ITEM PRICE"
@ 6,29 SAY "-----"
@ 8,14 SAY "This module is used when the item's final price"
@ 9,14 SAY "differs from that on the original requisition. You"
@ 10,14 SAY "will be shown all items associated with a specific"
@ 11,14 SAY "requisition # or purchase order #. Upon displaying"
@ 12,14 SAY "each item, you will be given the opportunity to"
@ 13,14 SAY "change the item's unit price or scroll to the next"
@ 14,14 SAY "item."
set color to &mscolor
@ 17,28 SAY "Continue (Y/N)?"
@ 17,45 GET ans
set color to &ccolor
read
if upper(ans) <> 'Y'
    return
endif
do while .T.
    ans1 = 3
    set scoreboard on
    clear
    @ 8,20 SAY "Preparing to change an item price"
    @ 16,32 SAY "Select:"
    @ 16,41 GET ans1 picture "#" range 1,3
    @ 17,22 SAY "-----"
    @ 18,22 SAY "[ 1. Show by requisition #   ]"
    @ 19,22 SAY "[ 2. Show by purchase order # ]"
    @ 20,22 SAY "[ 3. Return to main menu     ]"
    @ 21,22 SAY "-----"
    read
    set scoreboard off
    do case
        case ans1 = 1
            kind = "Requisition #"
            select D
            use
            use &item index &reqno_i
            field = "D->reqno"
        case ans1 = 2
            kind = "Purchase Order #"
            select D
            use
            use &item index &pono_i
            field = "D->pono"
        case ans1 = 3
            return
    endcase
    ans2 = ' '
    do while ans2 <> 'Y'

```

```

key = ' '
clear
@ 9,15 SAY "Enter"
@ 9,22 SAY kind
@ 9,38 SAY ":"
@ 9,41 GET key picture "*****"
@ 11,15 SAY "Note - Blank ends process"
read
if key = ' '
    return to master
endif
set color to &mscolor
@ 14,28 SAY "Correct (Y/N)?"
@ 14,44 GET ans2 picture "@!"
set color to &ccolor
read
enddo
find &key
if eof() .or. bof()
    @ 14,0 clear
    @ 14,23 SAY kind
    @ 14,39 SAY "not in file."
    @ 16,11 SAY "Please double check the"
    @ 16,35 SAY kind
    @ 16,53 SAY "and try again."
    set color to &mscolor
    wait &wmsg
    set color to &ccolor
else
    do while &field = key .and. .not. eof()
        ans3 = 'N'
        clear
        @ 7,11 SAY "Requisition #:"
        @ 7,27 SAY reqno
        @ 9,11 SAY "Purchase Ord #:"
        @ 9,28 SAY pono
        @ 11,11 SAY "Item:"
        @ 11,18 SAY descrip
        @ 13,11 SAY "Quantity:"
        @ 13,22 SAY quant
        @ 13,39 SAY "Unit Price: $"
        @ 13,53 SAY uprice
        set color to &mscolor
        @ 16,21 SAY "Change unit price (Y/N)?"
        @ 16,48 GET ans3
        set color to &ccolor
        read
        if upper(ans3) = 'Y'
            newprice = uprice
            @ 13,53 GET newprice picture "#####.##"
            read
            if newprice <> 0
                ans4 = ' '
                @ 16,0 clear
                set color to &mscolor
                @ 16,25 SAY "Correct (Y/N)?"
                @ 16,39 GET ans4
                set color to &ccolor
                read
                if upper(ans4) = 'Y'
                    replace uprice with newprice
                endif
            else
                ans4 = 'N'
                set color to &mscolor
                @ 21,21 SAY "Is item now FREE?"
                @ 21,46 GET ans4
                set color to &ccolor
                read
                if upper(ans4) = 'Y'

```

```
        replace uprice with newprice
      endif
    endif
  skip -1
endif
skip
enddo
endif
enddo
```

```

*****
*
* Module Name: CONFIG.PRG
*
* Author: Tom Trotter
*
* Purpose: After displaying an appropriate warning banner, this module
* displays the current program system configuration and allows
* the user to change it. Changes are saved to a file of
* memory variables (config.var).
*
* This module is called by: MAIN.PRG
*
* This module calls: N / A
*
*****

```

```

ans = 'N'
clear
@ 4,32 SAY "CONFIGURATION"
@ 5,32 SAY "-----"
@ 7,16 SAY "This program has been preconfigured for your use"
@ 8,16 SAY "and should not require reconfiguration unless"
@ 9,16 SAY "transported to another system or department."
@ 11,16 SAY "WARNING - Configuration changes should only be"
@ 12,26 SAY "made by personnel familiar with dBASE III."
set color to GB+/,W+/,
@ 15,23 SAY "Change Configuration (Y/N)?"
@ 15,52 GET ans
set color to GR+/,B,W+/,B+
read
if upper(ans) <> 'Y'
    return
endif
clear
restore from config.var
c_sdrv = substr(c_sdrv,1,1)
c_adrv = substr(c_adrv,1,1)
@ 2,17 SAY "This program is configured as follows:"
@ 6,17 SAY "Department:"
@ 6,30 SAY c_dept
@ 8,17 SAY "Bldg:"
@ 8,24 SAY c_bldg
@ 10,17 SAY "Room:"
@ 10,24 SAY c_room
@ 12,17 SAY "Telephone:"
@ 12,29 SAY c_phone
@ 14,17 SAY "Database disk locations: "
@ 15,37 SAY "Suppliers: ' ' drive"
@ 15,49 SAY c_sdrv
@ 17,37 SAY "Accounts: ' ' drive"
@ 17,49 SAY c_adrv
@ 19,17 SAY "Color Monitor (T/F):"
@ 19,38 SAY c_clrmon
ans = ' '
set color to R/,R/W
@ 21,17 SAY "Is this the correct configuration? (Y/N) "
@ 21,60 GET ans
set color to GR+/,B,W+/,B+
read
if upper(ans) = "N"
    ok = .F.
    do while .not. ok
        @ 6,30 GET c_dept
        @ 8,24 GET c_bldg
        @ 10,24 GET c_room
        @ 12,29 GET c_phone
        @ 15,49 GET c_sdrv

```

```

read
do while .not. (upper(c_sdrv) = 'A' .or. upper(c_sdrv) = 'B' .or. ;
  upper(c_sdrv) = 'C' .or. upper(c_sdrv) = 'D')
  @ 15,49 GET c_sdrv
  read
enddo
@ 17,49 GET c_adrv
read
do while .not. (upper(c_adrv) = 'A' .or. upper(c_adrv) = 'B' .or. ;
  upper(c_adrv) = 'C' .or. upper(c_adrv) = 'D')
  @ 17,49 GET c_adrv
  read
enddo
@ 19,38 GET c_clrmon
@ 21,60 GET ans
read
if upper(ans) = 'Y'
  ok = .T.
  c_sdrv = (c_sdrv) + ':'
  c_adrv = (c_adrv) + ':'
  save to config.var all like c_*
endif
enddo
endif
return

```



```

*****
*
* Module Name:  DISP_REQ.PRG
*
* Author: Tom Trotter
*
* Purpose:  This module presents the user with menus for use in the
*           selection of requisitions for display. The user is given
*           the option of screen or printer output.
*
* This module is called by:  MAIN.PRG
*
* This module calls:  DIS1_REQ
*                   DIS2_REQ
*                   DIS3_REQ
*
*****

```

```

do while .T.
  ans = ' '
  clear
  @ 7,28 SAY "DISPLAY REQUISITION"
  @ 8,28 SAY "-----"
  @ 10,15 SAY "This program will display one or more requisitions"
  @ 11,15 SAY "subject to user selection criteria. The list may"
  @ 12,15 SAY "be sent to the screen or printer at the user's"
  @ 13,15 SAY "option."
  set color to &mscolor
  @ 16,27 SAY "Continue (Y/N)?"
  @ 16,44 GET ans
  set color to &ccolor
  read
  if upper(ans) <> 'Y'
    return
  endif
  pick = 1
  clear
  @ 2,27 SAY "DISPLAY REQUISITIONS"
  set color to &mscolor
  @ 5,29 SAY "Selection:  "
  set color to &ccolor
  @ 5,41 GET pick picture "#" range 0,3
  @ 6,11 SAY "=====/"
  @ 7,11 SAY "/" 0. Return to main menu /"
  @ 8,11 SAY "/" 1. Display single requisition /"
  @ 9,11 SAY "/" 2. Display requisitions from specific account /"
  @ 10,11 SAY "/" 3. Display requisitions subject to account # & date /"
  @ 11,11 SAY "=====/"
  read
  if pick = 0
    return
  endif
  outpt = 1
  set color to &mscolor
  @ 14,27 SAY "Select Display:  "
  set color to &ccolor
  @ 14,44 GET outpt picture "#" range 1,2
  @ 15,27 SAY "=====/"
  @ 16,27 SAY "/" 1. Screen /"
  @ 17,27 SAY "/" 2. Printer /"
  @ 18,27 SAY "=====/"
  read
  public prtout
  if outpt = 1
    prtout = .F.
  else
    prtout = .T.
  endif
  do case

```

```
case pick = 1
  do dis1_req
case pick = 2
  do dis2_req
case pick = 3
  do dis3_req
endcase
enddo
```

```

*****
*
* Module Name:  DIS1_REQ.PRG
*
* Author: Tom Trotter
*
* Purpose:  This module outputs the contents of one requisition as
*           selected by the user input of a requisition #.
*
* This module is called by: DISP_REQ.PRG
*
* This module calls:  REQ_HDR.PRG
*                    WARN_PRT.PRG
*
*****

```

```

do while .T.
  clear
  key = ' '
  @ 9,25 SAY "Enter Requisition #:"
  @ 9,47 GET key picture "*****"
  @ 11,28 SAY "(Blank ends process.)"
  read
  if key = ' '
    return
  endif
  counter = 0
  marker = 9
  endif
  select C
  use
  use &req index &reqno_r
  find &key
  if eof()
    @ 14,28 SAY "REQUISITION NOT FOUND"
    @ 16,18 SAY "Double check the requisition # and try again."
  else
    totcnt = 0
    value = 0.00
    totval = 0.00
    if prtout
      .do warn_prt
      clear
      @ 10,31 SAY "LOAD PRINTER"
      @ 12,37 SAY "&"
      @ 14,25 SAY "PRESS ANY KEY TO CONTINUE"
      set device to print
      @ 1,0 SAY ' '
      wait ''
    endif
    key2 = C->accno
    select A
    use
    use &acct index &accno_a
    find &key2
    key3 = C->supno
    select B
    use
    use &sup index &supno_s
    find &key3
    do req_hdr
    select D
    use
    use &item index &reqno_i
    find &key
    do while D->reqno = C->reqno .and. .not. eof()
      counter = counter + 1
      marker = marker + 1
      totcnt = totcnt + D->quant
    enddo
  enddo
enddo

```

```

value = D->quant * D->uprice
totval = totval + value
@ marker,1 SAY counter picture "##"
@ marker,5 SAY D->descrip
@ marker,46 SAY "$"
@ marker,47 SAY D->uprice picture "##,##.##"
@ marker,57 SAY D->quant picture "#####"
@ marker,66 SAY D->rcvd
@ marker,70 SAY D->pono
if marker > 21 .and. .not. prtout
    marker = 9
    wait &wmsg
    @ 9,0 clear
endif
skip
enddo
if counter <> 0
    marker = marker + 3
    @ marker,15 SAY "Total items:"
    @ marker,28 SAY totcnt picture "#####"
    @ marker,42 SAY "Total Value: $"
    @ marker,57 SAY totval picture "##,##,##.##"
endif
if prtout
    @ marker,70 SAY chr(13)
    set device to screen
else
    set color to &mcolor
    wait &wmsg
    set color to &ccolor
endif
endif
enddo

```

```

*****
*
* Module Name:  DIS2_REQ
*
* Author: Tom Trotter
*
* Purpose:  This module displays all requisitions from a single account
*           as specified by the user.  The SEL_ACCNT module is used to
*           select an account, and this module will then skip through
*           the req.dbf and item.dbf until all requisitions for the
*           account have been displayed.
*
* This module is called by:  DISP_REQ.PRG
*
* This module calls:  SEL_ACCNT.PRG
*                   REQ_HDR.PRG
*                   WARN_PRT.PRG
*
*****

```

```

do sel_acnt
key = A->accno
select C
use
use &req index &accno_r
find &key
if eof()
  clear
  @ 15,18 SAY "No Requisitions on file for"
  @ 17,20 SAY "Account Name:"
  @ 17,34 SAY A->aname
  @ 19,20 SAY "Account Number:"
  @ 19,36 SAY A->accno
else
  clear
  if prtout
    do warn_prt
    clear
    @ 10,31 SAY "LOAD PRINTER"
    @ 12,37 SAY "&"
    @ 14,25 SAY "PRESS ANY KEY TO CONTINUE"
    set device to print
    @ 1,0 SAY ' '
    wait ''
  endif
  do while C->accno = A->accno .and. .not. eof()
    key2 = C->supno
    select B
    use
    use &sup index &supno_s
    find &key2
    counter = 0
    marker = 9
    totcnt = 0
    value = 0.00
    totval = 0.00
    do req_hdr
    key3 = C->reqno
    select D
    use
    use &item index &reqno_i
    find &key3
    endif
    do while D->reqno = C->reqno .and. .not. eof()
      counter = counter + 1
      marker = marker + 1
      totcnt = totcnt + quant
      value = D->quant * D->uprice
      totval = totval + value
    endwhile
  endwhile
endwhile

```



```

@ marker,1 SAY counter picture "###"
@ marker,5 SAY D->descrip
@ marker,46 SAY "$"
@ marker,47 SAY D->uprice
@ marker,57 SAY D->quant
@ marker,66 SAY D->rcvd
@ marker,70 SAY D->pono
if marker > 21 .and. .not. prtout
    marker = 9
    set color to &mscolor
    wait &wmsg
    set color to &ccolor
    @ 9,0 clear
endif
skip
enddo
if counter <> 0
    marker = marker + 3
    @ marker,15 SAY "Total items:"
    @ marker,28 SAY totcnt picture "#####"
    @ marker,42 SAY "Total value: $"
    @ marker,57 SAY totval picture "#,###,###.##"
endif
select C
skip
if .not. prtout
    set color to &mscolor
    wait &wmsg
    set color to &ccolor
endif
enddo
if prtout
    @ marker,70 SAY chr(13)
    set device to screen
endif
endif
return

```

```

*****
*
* Module Name:  DIS3_REQ.PRGM
*
* Author: Tom Trotter
*
* Purpose:  This module displays all requisitions from a user selected
*           account, subject to a screen on the origin date of the
*           requisition.  Output may be directed to the screen or the
*           printer at the user's option.
*
* This module is called by:  DISP_REQ.PRGM
*
* This module calls:  SEL_ACNT.PRGM
*                   REQ_HDR.PRGM
*                   WARN_PRT.PRGM
*
*****

```

```

hodate = today
lodate = CTOD("10/1/85")
clear
@ 6,26 SAY "-----"
@ 8,26 SAY "Find Requisitions Created"
@ 10,34 SAY "Between"
@ 12,26 GET lodate
@ 12,36 SAY "and"
@ 12,43 GET hodate
@ 13,26 SAY "mm/dd/yy"          mm/dd/yy"
read
do sel_acnt
counter = 0
marker = 9
key = A->accno
select C
use
use &req index &accno_r
find &key
if eof()
clear
@ 8,25 SAY "No Requisitions Are on File For"
@ 10,29 SAY "Account Name:"
@ 10,44 SAY A->aname
@ 12,29 SAY "Account Number:"
@ 12,46 SAY A->accno
@ 14,36 SAY "Between"
@ 16,27 SAY lodate
@ 16,38 SAY "and"
@ 16,46 SAY hodate
else
clear

if prtout
do warn_prt
clear
@ 10,31 SAY "LOAD PRINTER"
@ 12,37 SAY "&"
@ 14,25 SAY "PRESS ANY KEY TO CONTINUE"
set device to print
@ 1,0 SAY ' '
wait ''
endif
do while C->accno = A->accno .and. .not. eof()
if C->rdate >= lodate .and. C->rdate <= hodate
key2 = C->supno
select B
use
use &sup index &supno_s
find &key2

```

```

totcnt = 0
value = 0.00
totval = 0.00
counter = 0
marker = 9
do req_hdr
key3 = C->reqno
select D
use
use &item index &reqno_i
find &key3
do while D->reqno = C->reqno .and. .not. eof()
  counter = counter + 1
  marker = marker + 1
  @ marker,1 SAY counter picture "###"
  @ marker,5 SAY D->descrip
  @ marker,46 SAY "$"
  @ marker,47 SAY D->uprice
  @ marker,57 SAY D->quant
  @ marker,66 SAY D->rcvd
  @ marker,70 SAY D->pono
  totcnt = D->quant + totcnt
  value = D->quant * D->uprice
  totval = value + totval
  if marker > 21 .and. .not. prtout
    marker = 9
    set color to &mscolor
    wait &wmsg
    set color to &ccolor
    @ 9,0 clear
  endif
  skip
enddo
if counter <> 0
  marker = marker + 3
  @ marker,15 SAY "Total Items:"
  @ marker,28 SAY totcnt picture "#####"
  @ marker,45 SAY "Total Value: $"
  @ marker,60 SAY totval picture "$,###,###.##"
  set color to &mscolor
  wait &wmsg
  set color to &ccolor
endif
endif
select C
skip
enddo
if prtout
  @ marker,70 SAY chr(13)
  set device to screen
endif
if counter = 0
  clear
  @ 8,25 SAY "No Requisitions Are on File For"
  @ 10,29 SAY "Account Name:"
  @ 10,44 SAY A->aname
  @ 12,29 SAY "Account Number:"
  @ 12,46 SAY A->accno
  @ 14,36 SAY "Between"
  @ 16,27 SAY lodate
  @ 16,38 SAY "and"
  @ 16,46 SAY hidate
  set color to &mscolor
  wait &wmsg
  set color to &ccolor
endif
endif
return

```

```

*****
*
* Module Name:  FILE_ITM.PRG
*
* Author:  Tom Trotter
*
* Purpose:  This module is used to append a blank record to the item.dbf
*           and fill in several fields from current memory variables.
*
* This module is called by:  BUY2_ITM.PRG
*                           FORM_ORD.PRG
*                           PUB_ORD.PRG
*                           STK_ORDR.PRG
*
* This module calls:  N / A
*
*****

```

```

select D
use
use &item index &reqno_i,&pono_i
append blank
replace descrip with M->descrip
replace unit with M->unt
replace quant with M->qunt
replace uprice with M->uprce
replace reqno with M->reqno
use
return

```

```

*****
*
* Module Name:  FILE_REQ.PRG
*
* Author: Tom Trotter
*
* Purpose:  This module is used to append a blank record to the req.dbf
*           and fill in several fields from current memory variables.
*
* This module is called by:  XPOR_ORD.PRG
*                           FORM_ORD.PRG
*                           PUB_ORD.PRG
*                           STK_ORDR.PRG
*                           OPN_ORDR.PRG
*                           RSS_ORD.PRG
*
* This module calls:  N / A
*
*****

```

```

select C
use
use &req index &reqno_r,&accno_r
append blank
replace reqno with M->reqno
replace accno with M->accno
replace pri_code with M->pri
replace supno with M->supno
replace rdate with M->today
if typeordr <> 'XPORT'
  select A
  replace lastused with M->checker
endif
return

```



```

*****
*
* Module Name:  FORM_ORD.PRG
*
* Author: Tom Trotter
*
* Purpose:  This module coordinates the process of ordering forms
*           from NSC Oakland.
*
* This module is called by:  NEW_ORDR.PRG
*
* This module calls:  GET_RDD.PRG
*                   GET_CAT.PRG
*                   GET_PRI.PRG
*                   BUYI_ITM.PRG
*                   FILE_REQ.PRG
*                   FILE_ITM.PRG
*
*****

```

```

ans = ' '
clear
@ 8,33 SAY "FORMS  ORDER"
@ 9,33 SAY "-----"
@ 11,13 SAY "This program prepares the documents required to order"
@ 12,13 SAY "forms from NSC OAKLAND.  If you desire to order forms"
@ 13,13 SAY "from NAVPUBFORMCEN Philadelphia, do not continue."
set color to &mscolor
@ 16,30 SAY "Continue (Y/N)?"
set color to &ccolor
@ 16,47 GET ans
read
if upper(ans) <> 'Y'
    return
endif
public sendto,supno,route
sendto = '(00228) NSC OAKLAND, CA'
supno = '1111'
route = 'NOZ'
do get_rdd
do get_pri
do get_cat
do buyi_itm
public docid,signal,fund
docid = 'A0D'
signal = 'A'
if uprce = 0
    fund = 'Y6'
else
    fund = '2S'
endif
do prt1348a
do file_req
do file_itm
return to master

```

```

*****
*
* Module Name:  GET_CAT.PRG
*
* Author: Tom Trotter
*
* Purpose:  This module presents the user with a menu from which to
*           select the appropriate category of items to be purchased.
*           Other memory variable assignments are made based upon the
*           category selected by the user.  The user is also queried
*           concerning the recurring nature of the demand.
*
* This module is called by: FORM_ORD.PRG
*                           OPN_ORDR.PRG
*                           PUB_ORD.PRG
*                           STK_ORDR.PRG
*                           XPOR_ORD.PRG
*
* This module calls:  N / A
*
*****

```

```

public supadr,rmko
ans = ' '
if typeordr = 'OPEN' .or. typeordr = 'STOCK'
  do while ans <> 'Y'
    ans1 = 1
    clear
    @ 0,20 SAY "SELECT CATEGORY OF ITEMS TO BE PURCHASED"
    @ 1,20 SAY "-----"
    @ 3,11 SAY "1. CONSUMABLE SUPPLIES (e.g. subscriptions to periodicals"
    @ 4,17 SAY "cables, floppy disks, misc items for micro computers,"
    @ 5,17 SAY "and other items having a useful life of < 1 year)."

```

```

        rmko = 'P'
        supadr = 'P' + trim(A->accno) + 'MM'
    case ans1 = 5
        rmko = 'P'
        supadr = 'P' + trim(A->accno) + 'PM'
    case ans1 = 6
        rmko = 'Q'
        supadr = 'Q' + trim(A->accno) + 'QS'
    case ans1 = 7
        rmko = 'L'
        supadr = 'L' + trim(A->accno) + '00'
endcase
if A->research
    supadr = A->accno
endif
public demand
if typeordr = 'RSS$'
    demand = 'R'
else
    ans = ' '
    ans2 = ' '
    do while ans2 <> 'Y'
        clear
        @ 8,29 SAY "SPECIFY DEMAND TYPE"
        @ 9,29 SAY "-----"
        @ 11,18 SAY "Is there a recurring demand for the"
        @ 12,18 SAY "items/services on this requisition (Y/N)?"
        @ 12,62 GET ans picture "@!"
        read
        set color to &mcolor
        @ 15,30 SAY "Correct (Y/N)?"
        @ 15,46 GET ans2 picture "@!"
        set color to &ccolor
        read
        if ans = 'Y'
            demand = 'R'
        else
            demand = 'N'
        endif
    enddo
endif
return

```

```

*****
*
* Module Name:  GET_DATE.PRG
*
* Author:  Tom Trotter
*
* Purpose:  Since dBASE III does not allow one to change the system
*           date from within dBASE, this module allows the user to
*           update the memory var (TODAY) if it is in error. It also
*           will allow generation of documents dated other than the
*           current date. After the desired date is confirmed by the
*           user, a julian date is calculated. Leap years are accounted
*           for in the julian date calculation.
*
* This module is called by:  MAIN.PRG
*
* This module calls:  N / A
*
*****

```

```

clear memory
today = date()
do while .T.
  ans = ' '
  clear
  @ 5,28 SAY "CONFIRM TODAY'S DATE"
  @ 7,33 SAY today
  @ 9,31 SAY "(Y/N)?"
  @ 9,39 GET ans
  read
  if upper(ans) <> 'Y'
    clear
    @ 5,28 SAY "ENTER TODAY'S DATE"
    @ 7,33 GET today picture "dd"
    @ 8,33 SAY "MM/DD/YR"
    @ 15,10 SAY "Note - Leading zeros are required for single digit values."
    read
  else
    mon = month(today)
    do case
      case mon = 1
        julian = 0
      case mon = 2
        julian = 31
      case mon = 3
        julian = 59
      case mon = 4
        julian = 90
      case mon = 5
        julian = 120
      case mon = 6
        julian = 150
      case mon = 7
        julian = 181
      case mon = 8
        julian = 212
      case mon = 9
        julian = 243
      case mon = 10
        julian = 273
      case mon = 11
        julian = 304
      case mon = 12
        julian = 334
    endcase
    julian = julian + day(today)
    if int(year(today)/4) * 4 = year(today) .and. month(today) > 2
      julian = julian + 1
    endif
  endif

```

```
    save to date.var all except ans  
    return  
  endif  
enddo
```

```

*****
*
* Module Name:  GET_PRI.PRGM
*
* Author: Tom Trotter
*
* Purpose:  This module presents the user with a menu for the selection
*           of an order priority.  If the user selects pri "A" he is
*           warned that it requires approval by the NPS superintendent.
*           Other memory variables are assigned values for document
*           generation based upon the user's selection.
*
* This module is called by:  FORM.ORD.PRGM
*                           OPN_ORDR.PRGM
*                           PUB_ORD.PRGM
*                           STK_ORDR.PRGM
*                           XPOR_ORD.PRGM
*
* This module calls:  N / A
*
*****

```

```

public pri
ok = ' '
okl = ' '
pri = 'C'
do while .T.
  clear
  @ 4,30 SAY "ORDER PRIORITY"
  @ 6,12 SAY "A - Requirement is immediate and without the material"
  @ 7,16 SAY "required, the activity is unable to perform one"
  @ 8,16 SAY "or more of its primary missions."
  @ 9,17 SAY "(Note: Requires NPS Superintendent approval)"
  @ 11,12 SAY "B - Requirement is immediate or it is known that such"
  @ 12,16 SAY "a requirement will occur in the immediate future."
  @ 14,12 SAY "C - Routine requirement."
  @ 17,19 SAY "Select appropriate order priority:"
  @ 17,55 GET pri picture "a!"
  read
  set color to &mscolor
  @ 19,30 SAY "Correct (Y/N)?"
  @ 19,46 GET ok picture "a!"
  set color to &ccolor
  read
  if ok = 'Y' .and. pri <> 'A'
    okl = 'Y'
  endif
  if ok = 'Y' .and. pri = 'A'
    clear
    @ 7,34 SAY "NOTICE"
    @ 8,31 SAY "-----"
    @ 10,10 SAY "A priority code of 'A' requires NPS Superintendent approval."
    set color to &mscolor
    @ 14,18 SAY "Have you obtained approval (Y/N)?"
    @ 14,53 GET okl picture "a!"
    set color to &ccolor
    read
  endif
  if ok = 'Y' .and. okl = 'Y'
    public ms_code,priority
    good = .F.
    do case
      case pri = 'A'
        ms_code = 'W'
        priority = '08'
        good = .T.
      case pri = 'B'
        ms_code = 'T'
        priority = '10'
    endcase
  endif
endwhile

```



```
        good = .T.  
        case pri = 'C'  
            ms_code = 'T'  
            priority = '15'  
            good = .T.  
        endcase  
        if good  
            return  
        endif  
    endif  
enddo
```

```

*****
*
* Module Name:  GET_RDD.PRG
*
* Author: Tom Trotter
*
* Purpose:  This module queries the user for a "required delivery date."
*           It will accept RDD values from 1 to 366. If the user opts
*           to specify an RDD, RDD_YES will be given a value of .T., and
*           the memory variable RDD will be assigned a value (1-366).
*
* This module is called by: FORM_ORD.PRG
*                           OPN_ORDR.PRG
*                           PUB_ORD.PRG
*                           STK_ORDR.PRG
*                           XPOR_ORD.PRG
*
* This module calls:  N / A
*
*****

```

```

public rdd_yes,rdd
rdd = 1
ans = ' '
do while .T.
  rdd_yes = .F.
  clear
  @ 6,24 SAY "REQUIRED DELIVERY DATE (RDD)"
  @ 8,16 SAY "This is an optional 3 digit Julian date when the"
  @ 9,16 SAY "material or service is required. It is used to"
  @ 10,16 SAY "assist in the placement of the order."
  set color to &mcolor
  @ 12,22 SAY "Desire to specify an RDD (Y/N):"
  @ 12,55 GET ans
  set color to &ccolor
  read
  if upper(ans) = 'Y'
    set scoreboard on
    rdd_yes = .T.
    ok = ' '
    @ 14,30 SAY "Enter RDD:"
    @ 14,42 GET rdd picture "###" range 1,366
    read
    set scoreboard off
    @ 14,42 SAY rdd picture "###"
    set color to &mcolor
    @ 16,30 SAY "RDD Correct (Y/N):"
    @ 16,49 GET ok
    set color to &ccolor
    read
    if upper(ok) = 'Y'
      return
    endif
  else
    return
  endif
enddo

```

```

*****
*
*  Module Name: IN_PONO.PRG
*
*  Author: Tom Trotter
*
*  Purpose:  This module permits the user to write a purchase order
*            number into the PONO field of the ITEM.dbf.  This cannot
*            be done at the time the requisition is originated,
*            because the PONO is not then known.
*
*  This module is called by:  MAIN.PRG
*
*  This module calls:  N / A
*
*****

```

```

ans = 'Y'
clear
@ 6,23 SAY "INPUT PURCHASE ORDER NUMBER (PO#)"
@ 7,23 SAY "-----"
@ 9,11 SAY "Once a PO# is received from supply, this program permits"
@ 10,11 SAY "entry of the PO# for each item ordered.  Since not all"
@ 11,11 SAY "items on the original requisition may have been ordered"
@ 12,11 SAY "by supply on the same purchase order, you are requested"
@ 13,11 SAY "to separately confirm each item on the purchase order."
set color to &mscolor
@ 16,30 SAY "Continue (Y/N)?"
@ 16,47 GET ans
set color to &ccolor
read
if upper(ans) <> 'Y'
    return
endif
ans = ' '
do while .T.
    clear
    key = ' '
    @ 1,33 SAY "ENTER DATA"
    @ 2,33 SAY "-----"
    @ 3,22 SAY "Requisition # : "
    @ 3,39 GET key picture "@#####"
    @ 4,26 SAY "(Blank ends process)"
    read
    if key = ' '
        return
    endif
    tempid = ' '
    @ 6,22 SAY "Purchase Order #:"
    @ 6,41 GET tempid picture "@#####"
    set color to &mscolor
    @ 8,27 SAY "Correct (Y/N)?"
    @ 8,43 GET ans
    set color to &ccolor
    read
    if upper(ans) = 'Y'
        select D
        use
        use &item index &reqno_i,&pono_i
        find &key
        if eof() .or. bof()
            @ 8,0 clear
            @ 10,29 SAY "REQUISITION NOT IN FILE"
            @ 12,14 SAY "Please double check the requisition # and try again."
            set color to &mscolor
            wait &wmsg
            set color to &ccolor
        else
            do while D->reqno = key .and. .not. eof()

```

```

ans2 = ' '
clear
@ 7,18 SAY "Requisition #:"
@ 7,34 SAY D->reqno
@ 9,18 SAY "Item:"
@ 9,25 SAY D->descrip
@ 11,18 SAY "Quantity:"
@ 11,29 SAY D->quant
@ 13,18 SAY "Unit Price:"
@ 13,31 SAY D->uprice
set color to &mscolor
@ 18,14 SAY "Is this item on purchase #"
@ 18,42 SAY tempid
@ 18,53 SAY "(Y/N)?"
@ 18,61 GET ans2
set color to &ccolor
read
if upper(ans2) = 'Y'
    replace pono with M->tempid
endif
skip
enddo
endif
endif
enddo

```

```

*****
*
* Module Name:  IN_RCVD.PRG
*
* Author: Tom Trotter
*
* Purpose:  This module permits notation of receipt of items in the
*           RCVD field in the ITEM.dbf.  The item may be
*           specified by requisition # (REQNO) or purchase order #
*           (PONO)
*
* This module is called by:  MAIN.PRG
*
* This module calls:  N / A
*
*****

```

```

ans = 'Y'
clear
@ 3,30 SAY "INPUT RECEIPT STATUS"
@ 4,30 SAY "-----"
@ 6,16 SAY "You will be given the opportunity to indicate if"
@ 7,16 SAY "specific items, currently not marked as received,"
@ 8,16 SAY "have now been received.  Items reviewed will be"
@ 9,16 SAY "grouped by requisition number or purchase order"
@ 10,16 SAY "number at your option."
set color to &mscolor
@ 12,30 SAY "Continue (Y/N)?"
@ 12,47 GET ans
set color to &ccolor
read
if upper(ans) <> 'Y'
    return
endif
do while .T.
    clear
    set scoreboard on
    @ 7,22 SAY "PREPARING TO MARK ITEMS RECEIVED"
    ans1 = 1
    @ 16,32 SAY "Select:"
    @ 16,41 GET ans1 picture "#" range 1,3
    @ 17,22 SAY "-----"
    @ 18,22 SAY "[ 1. Show by requisition # ]"
    @ 19,22 SAY "[ 2. Show by purchase order # ]"
    @ 20,22 SAY "[ 3. Return to main menu ]"
    @ 21,22 SAY "-----"
    read
    set scoreboard off
    do case
        case ans1 = 1
            kind = "Requisition # "
            select D
            use
            use &item index &reqno_i
            field = "D->reqno"
        case ans1 = 2
            kind = "Purchase Order #"
            select D
            use
            use &item index &pono_i
            field = "D->pono"
        case ans1 = 3
            return
    endcase
    ans2 = 'Y'
    key = ' '
    clear
    @ 9,15 SAY "Enter"
    @ 9,22 SAY kind

```

```

@ 9,38 SAY ":"
@ 9,41 GET key picture "@#####"
@ 11,15 SAY "Note - Blank ends process"
read
if key = ' '
    return to master
endif
set color to &mscolor
@ 14,28 SAY "Correct (Y/N)?"
@ 14,44 GET ans2
set color to &ccolor
read
if upper(ans2) = 'Y'
    find &key
    if eof() .or. bof()
        @ 14,0 clear
        @ 14,24 SAY kind
        @ 14,40 SAY " not on file."
        @ 16,14 SAY "Please double check the "
        @ 16,38 SAY kind
        @ 16,54 SAY " and try again."
        set color to &mscolor
        wait &wmsg
        set color to &ccolor
        clear
    else
        counter = 0
        do while &field = key .and. .not. eof()
            if .not. rcvd
                counter = counter + 1
                ans3 = ' '
                clear
                @ 3,18 SAY "The following item shown as not received"
                @ 7,11 SAY "Requisition #:"
                @ 7,27 SAY reqno
                @ 9,11 SAY "Purchase Ord #:"
                @ 9,28 SAY pono
                @ 11,11 SAY "Item:"
                @ 11,18 SAY descrip
                @ 13,11 SAY "Quantity:"
                @ 13,22 SAY quant
                @ 13,39 SAY "Unit Price: $"
                @ 13,53 SAY uprice
                set color to &mscolor
                @ 19,21 SAY "Has this item been received (Y/N)?"
                @ 19,56 GET ans3
                set color to &ccolor
                read
                if upper(ans3) = 'Y'
                    replace rcvd with .T.
                endif
            endif
            skip
        enddo
        if counter = 0
            clear
            if ans1 = 1
                @ 12,18 SAY "All items for requisition #"
                @ 12,47 SAY &key
            else
                @ 12,16 SAY "All items for purchase order #"
                @ 12,47 SAY &key
            endif
            @ 14,18 SAY "have been previously marked as received."
            set color to &mscolor
            wait &wmsg
            set color to &ccolor
        endif
    endif
endif
endif

```


clear
enddo

```

*****
*
* Module Name:  MAIN.PRG
*
* Author: Tom Trotter
*
* Purpose:  Since this module is the first to be invoked, it is referred
*           to in dBASE III as the master module. It sets up the
*           initialization of key memory variables and presents the
*           main menu for program selection.
*
* This module is called by:  N / A
*
* This module calls:  WELCOME.PRG
*                   CONFIG.PRG
*                   GET_DATE.PRG
*                   NEW_ORDR.PRG
*                   IN_PONO.PRG
*                   IN_RCVD.PRG
*                   SHW_NRI.PRG
*                   DISP_REQ.PRG
*                   CHG_PRIC.PRG
*                   NEW_SUPL.PRG
*                   NEW_ACNT.PRG
*                   QUIT.PRG
*
*****

```

```

set help off
set talk off
set delete on
set safety off
set console on
set escape off
do welcome
do config
do get_date
do while .T.
  clear all
  public today,julian,wmsg
  wmsg = ""
                                     Press any key to continue..."
  public sup,sname_s,supno_s, template,tempord
  public item,reqno_i,pono_i,req,reqno_r,accno_r,acct,aname_a,accno_a
  restore from date.var additive
  public c_dept,c_bldg,c_room,c_phone
  restore from config.var additive
  sup = c_sdrv + "sup"
  sname_s = c_sdrv + "sname_s"
  supno_s = c_sdrv + "supno_s"
  template = c_adrv + "template"
  tempord = c_adrv + "tempord.dbf"
  item = c_adrv + "item"
  reqno_i = c_adrv + "reqno_i"
  pono_i = c_adrv + "pono_i"
  req = c_adrv + "req"
  reqno_r = c_adrv + "reqno_r"
  accno_r = c_adrv + "accno_r"
  acct = c_adrv + "acct"
  aname_a = c_adrv + "aname_a"
  accno_a = c_adrv + "accno_a"
  fy_id = c_adrv + "fy.id"
  public fy
  restore from &fy_id additive
  public ccolor,mscolor,errcolor
  if c_clrmon
    store "GR+/B,W+/ ,B+" to ccolor
    store "GR+/ ,GR+/ " to mscolor
    store "R+" to errcolor

```

```

else
  store "W+/ " to ccolor
  store "W+/ " to mscolor
  store "W+/ " to errcolor
endif
set color to &ccolor
choice = 0
set scoreboard on
clear
@ 3,10 SAY "=====
@ 4,10 SAY "=
@ 5,10 SAY "=====
@ 6,10 SAY "SUPPLY REQUISITION DATABASE"
@ 7,10 SAY "=====
@ 8,10 SAY "=
@ 9,10 SAY "0. Exit"
@ 10,10 SAY "1. Place an order"
@ 11,10 SAY "2. Update files with purchase order #"
@ 12,10 SAY "3. Mark items received"
@ 13,10 SAY "4. List items not yet received"
@ 14,10 SAY "5. Display a specific requisition"
@ 15,10 SAY "6. Change an item price"
@ 16,10 SAY "7. Enter a new supplier"
@ 17,10 SAY "8. Create a new account"
@ 18,10 SAY "=
@ 19,10 SAY "=====Your choice > <=====
@19,41 GET choice picture '#' range 0,8
read
set scoreboard off
do case
  case choice = 0
    do quit
  case choice = 1
    do new_ordr
  case choice = 2
    do in_pono
  case choice = 3
    do in_rcvd
  case choice = 4
    do shw_nri
  case choice = 5
    do disp_req
  case choice = 6
    do chg_pric
  case choice = 7
    do new_supl
  case choice = 8
    do new_acnt
endcase
enddo

```

```

*****
*
* Module Name:  NEW_ACNT.PRG
*
* Author:  Tom Trotter
*
* Purpose:  This module allows the user to set up a new account by
*           appending a blank record to the ACCNT.dbf and filling
*           in all fields.
*
* This module is called by:  MAIN.PRG
*                           SEL_ACNT
*
* This module calls:  N / A
*
*****

```

```

ans = ' '
do while ans <> 'Y'
  clear
  @ 3,28 SAY "CREATE A NEW ACCOUNT"
  @ 4,28 SAY "-----"
  @ 6,14 SAY "In order to enter a new account you must have"
  @ 7,14 SAY "the following information available at this time:"
  @ 9,17 SAY "Account Name"
  @ 10,17 SAY "Account # (assigned by NPS Comptroller)"
  @ 11,17 SAY "Serial # range assigned to account"
  @ 12,17 SAY "Is it a research account?"
  @ 14,14 SAY "WARNING - If you do not understand any of the"
  @ 15,24 SAY "above items, seek assistance and do"
  @ 16,24 SAY "not continue further at this time."
  @ 19,27 SAY "Continue (Y/N)?"
  @ 19,44 GET ans picture "a!"
  read
  if upper(ans) = 'N'
    return
  endif
enddo
select A
use
use &acct index &aname_a,&accno_a
append blank
do while .T.
  ans = ' '
  clear
  @ 1,25 SAY "INPUT NEW ACCOUNT"
  @ 2,25 SAY "-----"
  @ 5,22 SAY "Account Name:"
  @ 5,37 GET aname picture "a!"
  @ 7,22 SAY "Account Number:"
  @ 7,39 GET accno picture "a!"
  @ 9,22 SAY "Code:"
  @ 9,29 GET code picture "a!"
  @ 11,22 SAY "Research Account (T/F):"
  @ 11,47 GET research picture "L"
  @ 13,22 SAY "Requisition Serial # Range - From:"
  @ 13,59 GET loserno picture "9999"
  @ 14,52 SAY "To:"
  @ 14,59 GET hiserno picture "9999"
  @ 15,22 SAY "Last Serial # Used:"
  @ 15,43 GET lastused picture "####"
  @ 16,22 SAY "(leave blank if account never used yet)"
  read
  if lastused = ' '
    replace lastused with loserno
  endif
  @ 19,27 SAY "Correct (Y/N)?"
  @ 19,43 GET ans
  read

```

```
    if upper(ans) = 'Y'  
        return  
    endif  
enddo
```

```

*****
*
* Module Name:  NEW_ORDR.PRG
*
* Author:  Tom Trotter
*
* Purpose:  This module serves as the menu driver for the activities
*           required in the generation of new orders.  It assigns a
*           value to the public memory variable TYPEORDR based upon
*           the user's selection.
*
* This module is called by:  MAIN.PRG
*
* This module calls:  CALC_REQ.PRG
*                   FORM_ORD.PRG
*                   OPN_ORD.PRG
*                   PUB_ORD.PRG
*                   RSS_ORD.PRG
*                   SEL_ACNT.PRG
*                   STK_ORD.PRG
*                   XPOR_ORD.PRG
*
*****

```

```

do while .T.
  order = 7
  clear
  set scoreboard on
  @ 3,29 SAY "PLACE AN ORDER"
  @ 5,10 SAY "The purpose of this program is to generate the appropri"
  @ 5,65 SAY "ate"
  @ 6,10 SAY "supply forms (DD 1348-6 &/or SF 36) to place an order."
  @ 9,24 SAY "Select the type of order"
  @ 11,10 SAY "1.  Open Purchase"
  @ 12,10 SAY "2.  Standard stock items from NPS RSS, NSC Oakland, or GSA"
  @ 13,10 SAY "3.  Standard forms from NSC Oakland"
  @ 14,10 SAY "4.  Publications from NAVPUBFORMCEN Phil"
  @ 15,10 SAY "5.  $$ Reimbursement to NPS Ready Supply Store"
  @ 16,10 SAY "6.  Transportation of material (e.g. Federal Express)"
  @ 17,10 SAY "7.  Return to main menu"
  @ 20,26 SAY "Your order:"
  @ 20,41 GET order picture "#" range 1,7
  read
  set scoreboard off
  public typeordr,service
  service = substr(fy,2,1)
  typeordr = ' '
  if order <> 6 .and. order <> 7
    do sel_acnt
    do calc_req
  endif
  do case
    case order = 1
      typeordr = 'OPEN'
      do opn_ordr
    case order = 2
      typeordr = 'STOCK'
      do stk_ordr
    case order = 3
      typeordr = 'FORM'
      do form_ord
    case order = 4
      typeordr = 'PUB'
      do pub_ord
    case order = 5
      typeordr = 'RSS$'
      do rss_ord
    case order = 6
      typeordr = 'XPORT'

```



```
do sel_acnt  
do calc_req  
do xpor_ord  
case order = 7  
return to master  
endcase  
enddo
```

```

*****
*
* Module Name:  NEW_SUPL.PRG
*
* Author: Tom Trotter
*
* Purpose:  This module allows the user to enter a new supplier by
*           appending a blank record to the SUP.dbf and filling in
*           all fields.
*
* This module is called by:  MAIN.PRG
*                           SEL_SUPL.PRG
*
* This module calls:  N / A
*
*****

```

```

ans = ' '
do while ans <> 'Y'
  clear
  @ 4,26 SAY "INPUT A NEW SUPPLIER"
  @ 5,26 SAY "-----"
  @ 7,12 SAY "In order to input a new supplier, you must have the"
  @ 8,12 SAY "following information available at this time:"
  @ 10,29 SAY "Supplier Name"
  @ 11,29 SAY "Address"
  @ 12,29 SAY "Phone #"
  @ 14,12 SAY "The phone number is mandatory. Do not proceed unless"
  @ 15,12 SAY "you are ready to enter the above information."
  @ 18,25 SAY "Continue (Y/N) ?"
  @ 18,43 GET ans picture "@!"
  read
  if ans = 'N'
    return
  endif
enddo
select B
use
use &sup index &supno_s,&sname_s
append blank
do while .T.
  ans = ' '
  clear
  @ 3,27 SAY "INPUT NEW SUPPLIER"
  @ 4,27 SAY "-----"
  @ 6,15 SAY "Name:"
  @ 6,23 GET sname
  @ 8,15 SAY "Address:"
  @ 8,25 GET addr
  @ 10,15 SAY "City:"
  @ 10,22 GET city
  @ 10,38 SAY "State:"
  @ 10,46 GET state picture "@!"
  @ 10,52 SAY "Zip:"
  @ 10,58 GET zip
  @ 12,15 SAY "Phone: (   )"
  @ 12,23 GET areacode picture "@###"
  @ 12,28 GET fone picture "@###-####"
  @ 14,15 SAY "Note - Phone # is mandatory"
  read
  @ 16,26 SAY "Is this correct?"
  @ 16,44 GET ans picture "@!"
  read
  tempid = substr(fone,4)
  if val(tempid) > 9970
    tempid = '0000'
  endif
  new = recno()
  if ans = 'Y'

```

```

tempid = str(val(tempid) + 1,4)
go top
find &tempid
if eof () .or. bof ()
    go new
    replace supno with tempid
    @ 14,0 clear
    @ 16,24 SAY "Assigned Supplier #:"
    @ 16,46 SAY supno
    wait &wmsg
    return
endif
endif
enddo

```

```

*****
*
* Module Name:  OPN_ORDR.PRG
*
* Author: Tom Trotter
*
* Purpose:  This module coordinates the activities necessary to make an
* open purchase.  It invokes several subordinate modules to
* gather various user inputs.  If only one item is being ordered
* and the description of that item will fit in the space
* allowed on a DD-1348 ( < 23 characters), then a DD-1348 will
* be generated.  If more than one item is being ordered, or if
* the item's description is lengthy, a SF-36 will be prepared
* in addition to a DD-1348.
*
* This module is called by:  MAIN.PRG
*
* This module calls:  SEL_SUPL.PRG
*                     GET_RDD.PRG
*                     GET_PRI.PRG
*                     GET_CAT.PRG
*                     BUY2_ITM.PRG
*                     PRTSF36.PRG
*                     PRT1348o.PRG
*                     SUP_MEMO.PRG
*                     FILE_REQ.PRG
*
*****

```

```

ans = ' '
clear
@ 3,27 SAY "OPEN PURCHASE REQUISITION"
@ 4,27 SAY "-----"
@ 6,12 SAY "This program prepares the necessary documents to execute"
@ 7,12 SAY "an open purchase request. A DOD 1348 (6 part) will always"
@ 8,12 SAY "be produced, and if the item description is too long"
@ 9,12 SAY "(greater than 22 characters), or if there is more than"
@ 10,12 SAY "one item being ordered, a SF 36 will also be prepared."
@ 11,12 SAY "If the total order value exceeds $1,000., a memorandum"
@ 12,12 SAY "will be prepared listing at least 2 alternative sources,"
@ 13,12 SAY "unless this procurement request is being accompanied by"
@ 14,12 SAY "a sole-source justification. A maximum of 10 separate"
@ 15,12 SAY "line items may be ordered on a single requisition."
@ 17,30 SAY "Continue (Y/N)?"
@ 17,47 GET ans
read
if upper(ans) <> 'Y'
    return
endif
do sel_supl
do get_rdd
do get_pri
do get_cat
public counter
counter = 1
do buy2_itm
public docid,signal,fund,sf36
docid = 'A0E'
signal = 'A'
fund = '2S'
select E
use &tempord
go bottom
if recno() = 1 .and. len(trim(E->descrip)) < 23 .and. E->descrip2 = ' '
    sf36 = .F.
else
    sf36 = .T.
    do prtsf36
endif

```

```
do prtl348o
do file_req
if totval > 999.99
  do sup_memo
endif
select E
use
erase &tempord
return to master
```

```

*****
*
* Module Name:  PRNT_SMO.PRG
*
* Author:  Tom Trotter
*
* Purpose:  This module prepares a supplemental memo to NPS Supply
*           memo to accompany an open purchase requisition if the total
*           value of all items on the requisition exceeds $1,000. The
*           purpose of the memo is to offer 2 alternative sources for
*           the requested material as required by Naval Supply regulations,
*           except in the case of orders accompanied by a sole source
*           justification.
*
* This module is called by:  SUP_MEMO.PRG
*
* This module calls:  N / A
*
*****

```

```

@ 1,7 SAY "From:"
@ 1,14 SAY c_dept
@ 1,18 SAY "Department"
@ 1,56 SAY today
@ 3,7 SAY "To: NPS Supply"
@ 5,7 SAY "Subj: Additional sources for requisition #"
@ 5,50 SAY reqno
@ 7,7 SAY "1. Due to the high value of subject requisition, the f"
@ 7,62 SAY "ollowing"
@ 8,7 SAY "multiple supply sources are submitted:"
@ 12,17 SAY sname1
@ 12,45 SAY "Phone:"
@ 12,53 SAY areacode1
@ 12,57 SAY fone1
@ 13,17 SAY addr1
@ 14,17 SAY city1
@ 14,37 SAY statel
@ 14,44 SAY zip1
@ 17,17 SAY sname2
@ 17,45 SAY "Phone:"
@ 17,53 SAY areacode2
@ 17,57 SAY fone2
@ 18,17 SAY addr2
@ 19,17 SAY city2
@ 19,37 SAY state2
@ 19,44 SAY zip2
@ 23,17 SAY sname3
@ 23,45 SAY "Phone:"
@ 23,53 SAY areacode3
@ 23,57 SAY fone3
@ 24,17 SAY addr3
@ 25,17 SAY city3
@ 25,37 SAY state3
@ 25,44 SAY zip3
@ 29,7 SAY "2. Department points of contact for this request are:"
@ 31,23 SAY poc1
@ 31,48 SAY ext1
@ 33,23 SAY poc2
@ 33,48 SAY ext2
@ 34,1 SAY chr(12)
set device to screen
return

```



```

*****
*
* Module Name:  PRTSF36.PRG
*
* Author:  Tom Trotter
*
* Purpose:  This module contains loop which will read one item at a time
*           from the TEMPORD.dbf and generate an SF-36.  The TEMPORD.dbf
*           is used as the information source because it contains the
*           fields of DESCRIP2 and DESCRIP3 which are not permanently
*           filed in the ITEM.dbf.  This allows a fuller description to
*           be used in the generation of the documents, yet provides for
*           a more economical filing structure for long term files.  Once
*           the user states that an acceptable SF-36 has been generated,
*           the TEMPORD.dbf is erased.  This module will produce an SF-36
*           containing from 1 to 12 line items.
*
* This module is called by:  OPN_ORDR.PRG
*
* This module calls:  WARN_PRT.PRG
*
*****

```

```

do warn_prt
ans = '-'
do while ans <> 'N'
  clear
  @ 10,26 SAY "LOAD SF-36 INTO PRINTER"
  @ 12,37 SAY "&"
  @ 14,25 SAY "PRESS ANY KEY TO CONTINUE"
  set device to print
  @ 1,0 SAY ' '
  wait ''
  req1 = substr(reqno,1,4)
  req2 = substr(reqno,5,4)
  ref = "N62271/" + req1 + '/' + req2
  @ 1,51 SAY ref
  @ 1,72 SAY "1 1"
  who = trim(B->sname)
  where = trim(B->addr) + ', ' + trim(B->city) + ', ' + trim(B->state) ;
         + ' ' + (B->zip)
  who = who + ', ' + where
  @ 3,1 SAY who
  select E
  go top
  marker = 7
  do while .not. eof()
    @ marker,3 SAY E->itemno
    @ marker,8 SAY E->descrip
    @ marker,49 SAY E->quant picture "#####"
    @ marker,57 SAY E->unit
    @ marker,61 SAY E->uprice picture "#####.##"
    amount = E->quant * E->uprice
    @ marker,71 SAY amount picture "#####.##"
    if descrip2 <> ' '
      marker = marker + 1
      @ marker,8 SAY E->descrip2
    endif
    if descrip3 <> ' '
      marker = marker + 1
      @ marker,8 SAY E->descrip3
    endif
    marker = marker + 2
    skip
  enddo
  @ marker,70 SAY "_____"
  marker = marker + 1
  @ marker,35 SAY "TOTAL"
  @ marker,70 SAY "$"

```

```
@ marker,71 SAY totval picture "##,##.##"  
@ 16,1 SAY chr(13)  
set device to screen  
clear  
@ 12,21 SAY "Do you want to reprint the SF-36?"  
@ 12,56 GET ans picture "@!"  
read  
enddo  
return
```

```

*****
*
* Module Name: PRT1348a.PRG
*
* Author: Tom Trotter
*
* Purpose: This module contains a loop which will print memory variables
*          to the appropriate spaces on a DD-1348. It is designed to
*          handle slight variations in format based upon the value of
*          TYPEORDR.
*
* This module is called by: STK_ORDR.PRG
*                          FORM_ORD.PRG
*                          PUB_ORD.ORG
*                          RSS_ORD.PRG
*
* This module calls: WARN_PRT.PRG
*
*****

```

```

do warn_prt
ans = ' '
do while ans <> 'N'
  clear
  @ 10,31 SAY "LOAD PRINTER"
  @ 12,37 SAY "&"
  @ 14,25 SAY "PRESS ANY KEY TO CONTINUE"
  set device to print
  @ 1,0 SAY ' '
  wait ' '
  @ 1,6 SAY sendto
  @ 1,36 SAY "(62271) NPS,"
  dept = trim(c_dept) + ' Dept,'
  @ 1,49 SAY dept
  ccode = 'Code ' + trim(A->code)
  @ 1,59 SAY ccode
  @ 2,36 SAY "Rm"
  @ 2,39 SAY c_room
  @ 2,45 SAY "Bldg"
  @ 2,50 SAY c_bldg
  @ 2,56 SAY c_phone
  if typeordr = 'STOCK'
    @ 4,1 SAY nomen1
    @ 5,1 SAY nomen2
    @ 5,25 SAY docid
  endif
  @ 5,30 SAY route
  if typeordr = 'RSS$'
    @ 5,38 SAY money
  else
    @ 5,34 SAY ms_code
    snum1 = substr(stknum,1,4)
    snum2 = substr(stknum,5,2)
    snum3 = substr(stknum,7,3)
    snum4 = substr(stknum,10,4)
    snum = snum1 + ' ' + snum2 + ' ' + snum3 + ' ' + snum4
    @ 5,37 SAY snum
    @ 5,59 SAY unt
    @ 5,63 SAY qunt picture "#####"
  endif
  @ 8,1 SAY "X 62271"
  req1 = substr(reqno,1,4)
  req2 = substr(reqno,5,4)
  req3 = req1 + ' ' + req2
  @ 8,11 SAY req3
  @ 8,23 SAY demand
  @ 8,25 SAY service
  @ 8,27 SAY supadr
  @ 8,34 SAY signal

```

```

if typeodr = 'FORM' .or. typeodr = 'PUB'
  @ 8,38 SAY nomen1
endif
@ 8,64 SAY rmko
@ 8,66 SAY "2D"
if typeodr = 'FORM' .or. typeodr = 'PUB'
  @ 9,38 SAY nomen2
endif
@ 12,1 SAY fund
@ 12,5 SAY dist
@ 12,9 SAY "OP4"
@ 12,14 SAY priority
if rdd_yes
  @ 12,17 SAY rdd picture "###"
endif
@ 15,4 SAY "AA"
do case
  case typeodr = 'RSS$'
    @ 15,49 SAY uprce picture "#####.##"
  case typeodr = 'PUB'
    @ 15,50 SAY "N/C"
  otherwise
    @ 15,33 SAY uprce picture "#####.##"
    totval = qunt * uprce
    @ 15,49 SAY totval picture "##,###.##"
  endcase
@ 16,1 SAY chr(13)
set device to screen
clear
@ 12,20 SAY "Do you want to reprint the DD-1348?"
@ 12,56 GET ans picture "@"
read
enddo
return

```

```

*****
*
* Module Name:  PRT1348o.PRG
*
* Author: Tom Trotter
*
* Purpose:  This module prints a DD-1348 for open order purchases. It is
*           designed to generate a DD-1348 which may be accompanied by
*           an SF-36, or one which can stand alone, depending upon the
*           value of the logical variable SF36.
*
* This module is called by:  OPN_ORDR.PRG
*
* This module calls:  WARN_PRT.PRG
*
*****

```

```

do warn_prt
ans = ' '
do while ans <> 'N'
  clear
  @ 10,31 SAY "LOAD PRINTER"
  @ 12,37 SAY "&"
  @ 14,25 SAY "PRESS ANY KEY TO CONTINUE"
  set device to print
  @ 1,0 SAY ' '
  wait ' '
  @ 1,1 SAY B->sname
  @ 1,36 SAY "(62271) NPS,"
  dept = trim(c_dept) + ' Dept,'
  @ 1,49 SAY dept
  ccode = 'Code ' + trim(A->code)
  @ 1,59 SAY ccode
  @ 2,1 SAY where
  @ 2,36 SAY "Rm"
  @ 2,39 SAY c_room
  @ 2,45 SAY "Bldg"
  @ 2,50 SAY c_bldg
  @ 2,56 SAY c_phone
  @ 5,25 SAY docid
  @ 5,34 SAY ms_code
  if sf36
    @ 5,38 SAY "SEE ATTACHED SF-36"
  else
    @ 5,38 SAY E->descrip
    @ 5,59 SAY E->unit
    @ 5,63 SAY E->quant picture "#####"
  endif
  @ 8,1 SAY "X 62271"
  req1 = substr(reqno,1,4)
  req2 = substr(reqno,5,4)
  req3 = req1 + ' ' + req2
  @ 8,11 SAY req3
  @ 8,23 SAY demand
  @ 8,25 SAY service
  @ 8,27 SAY supadr
  @ 8,34 SAY signal
  @ 8,64 SAY rmko
  @ 8,66 SAY "2D"
  @ 12,1 SAY fund
  @ 12,5 SAY dist
  @ 12,9 SAY "OP4"
  @ 12,14 SAY priority
  if rdd_yes
    @ 12,17 SAY rdd picture "###"
  endif
  @ 15,4 SAY "AA"
  if .not. SF36
    @ 15,33 SAY uprce picture "#####.##"
  endif

```

```
endif
@ 15,49 SAY totval picture "##,###.##"
@ 16,1 SAY chr(13)
set device to screen
clear
@ 12,20 SAY "Do you want to reprint the DD-1348?"
@ 12,56 GET ans picture "@!"
read
enddo
return
```



```

*****
*
* Module Name:  PRT1348x.PRG
*
* Author: Tom Trotter
*
* Purpose:  This module prints a DD-1348 for transportation requisitions.
*           It is formatted to include a field showing to whom the item
*           is to be sent.
*
* This module is called by:  XPOR_ORD.PRG
*
* This module calls:  WARN_PRT.PRG
*
*****

```

```

do warn_prt
ans = ' '
do while ans <> 'N'
  clear
  @ 10,31 SAY "LOAD PRINTER"
  @ 12,37 SAY "8"
  @ 14,25 SAY "PRESS ANY KEY TO CONTINUE"
  set device to print
  @ 1,0 SAY ' '
  wait ''
  select B
  use
  use &sup index &supno_s
  key = M->supno
  find &key
  sendto = B->sname
  @ 1,6 SAY sendto
  @ 1,36 SAY "(62271) NPS,"
  dept = trim(c_dept) + ' Dept,'
  @ 1,49 SAY dept
  ccode = 'Code ' + trim(A->code)
  @ 1,59 SAY ccode
  @ 2,36 SAY "Rm"
  @ 2,39 SAY c_room
  @ 2,45 SAY "Bldg"
  @ 2,50 SAY c_bldg
  @ 2,56 SAY c_phone
  @ 5,25 SAY docid
  @ 5,38 SAY "Ticket to:"
  @ 5,59 SAY unt
  @ 7,38 SAY shipto
  @ 8,1 SAY "X 62271"
  req1 = substr(reqno,1,4)
  req2 = substr(reqno,5,4)
  req3 = req1 + ' ' + req2
  @ 8,11 SAY req3
  @ 8,23 SAY demand
  @ 8,25 SAY service
  @ 8,27 SAY supadr
  @ 8,34 SAY signal
  @ 8,38 SAY addr1
  @ 8,64 SAY rmko
  @ 8,66 SAY "2D"
  @ 9,38 SAY addr2
  @ 10,38 SAY addr3
  @ 11,38 SAY addr4
  @ 12,1 SAY fund
  @ 12,9 SAY "OP4"
  @ 12,14 SAY priority
  @ 15,4 SAY "AA"
  @ 15,50 SAY uprce picture "####.##"
  @ 16,1 SAY chr(13)
  set device to screen

```

```
clear
@ 12,20 SAY "Do you want to reprint the DD-1348?"
@ 12,56 GET ans picture "@!"
read
enddo
return
```

```

*****
*
* Module Name:  PUB_ORD.PRG
*
* Author:  Tom Trotter
*
* Purpose:  This module coordinates the process of ordering forms
*           from NAVPUBFORMCEN, Philadelphia, PA.
*
* This module is called by:  NEW_ORDR.PRG
*
* This module calls:  GET_RDD.PRG
*                   GET_CAT.PRG
*                   GET_PRI.PRG
*                   BUY1_ITM.PRG
*                   FILE_REQ.PRG
*                   FILE_ITM.PRG
*
*****

```

```

ans = ' '
clear
@ 8,31 SAY "PUBLICATION ORDER"
@ 9,31 SAY "-----"
@ 11,13 SAY "This program prepares the documents required to order"
@ 12,13 SAY "publications from NAVPUBFORMCEN, Philadelphia. "
@ 16,30 SAY "Continue (Y/N)?"
@ 16,47 GET ans
read
if upper(ans) <> 'Y'
    return
endif
public sendto,supno,route
sendto = '(00288) NAVPUBFORMCEN PHILA'
supno = '2222'
route = 'NFZ'
do get_rdd
do get_pri
do get_cat
do buy1_itm
public docid,signal,fund
docid = 'A0D'
signal = 'D'
if uprce = 0
    fund = 'Y6'
else
    fund = '2S'
endif
do prt1348a
do file_req
do file_itm
return to master

```

```

*****
*
* Module Name:      QUIT.PRG
*
* Author: Norm Lyons 10/85
* Modified by: Tom Trotter 12/85 (converted to dBASE III)
*
* Purpose      : This program terminates processing and returns
*                control to the operating system. At the end of
*                each appropriate menu, an option to to quit is
*                displayed. If the user enters "Q" to quit, the
*                program ends and control is returned to the
*                operating system.
*
* This module is called by:  MAIN.PRG
*
* This module calls:  N / A
*
*****

```

```

@ 5,28 SAY "End of Dialog"
@ 7,10 SAY "If you are using floppy disks, replace them in their"
@ 9,10 SAY "protective jackets. Do not touch the surface of"
@ 11,10 SAY "the disk (the little oval opening at the bottom of the"
@ 13,10 SAY "disk)."
set color to &mscolor
@ 22,9 SAY "Press any key to return control to the operating system"
set color to &ccolor
wait ''
quit

```

```

*****
*
* Module Name:  REQ_HDR.PRG
*
* Author: Tom Trotter
*
* Purpose:  This module is used by the various modules which display
*           requisitions to put a common header on the screen.  It
*           has been put into a separate module in order to avoid
*           code duplication.
*
* This module is called by:  DIS1_REQ.PRG
*                           DIS2_REQ.PRG
*                           DIS3_REQ.PRG
*
* This module calls:  N / A
*
*****

```

```

clear
@ 1,5 SAY "Account Name:"
@ 1,19 SAY A->aname
@ 1,51 SAY "Requisition #:"
@ 1,66 SAY C->reqno
@ 2,5 SAY "Account #:"
@ 2,16 SAY C->accno
@ 2,51 SAY "Priority:"
@ 2,61 SAY C->pri_code
@ 3,21 SAY "To:"
@ 3,25 SAY B->sname
@ 3,51 SAY "Date:"
@ 3,57 SAY C->rdate
@ 4,25 SAY B->addr
where = trim(B->city) + ', ' + B->state + ' ' + B->zip
@ 5,25 SAY where
@ 7,2 SAY " #           Item                      Unit Cost"
@ 7,58 SAY "Quant  Rcvd  P.O. #"
@ 8,1 SAY "-----"
@ 8,56 SAY "-----"
return

```

```

*****
*
* Module Name:  RSS_ORD.PRG
*
* Author:  Tom Trotter
*
* Purpose:  This module prepares a DD-1348 for "Money Value Only" as a
*           method of payment in response to regular billings from the
*           NPS Ready Supply Store (RSS).
*
* This module is called by:  NEW_ORDR.PRG
*
* This module calls:  GET_CAT.PRG
*                   FILE_REQ.PRG
*                   PRT1348a.PRG
*
*****

```

```

ans = ' '
clear
@ 8,31 SAY "$$ REIMBURSEMENT TO RSS"
@ 9,31 SAY "-----"
@ 11,13 SAY 'This program prepares a DD-1348 for "money value only"'
@ 12,13 SAY "as reimbursement to the NPS RSS."
@ 16,30 SAY "Continue (Y/N)?"
@ 16,47 GET ans
read
if upper(ans) <> 'Y'
    return
endif
public dist,uprce
dist = ' '
uprce = 0.00
ans = ' '
clear
do while ans <> 'Y'
    @ 6,32 SAY "RSS $$ REIMBURSEMENT"
    @ 7,32 SAY "-----"
    @ 10,33 SAY "$$ Amount: $"
    @ 10,46 GET uprce picture "####.##"
    @ 13,29 SAY "COG Symbol on billing:"
    @ 13,53 GET dist picture "@!"
    read
    @ 16,32 SAY "Correct (Y/N)?"
    @ 16,48 GET ans picture "@!"
    read
enddo
public sendto,supno,route,money,priority,rdd_yes,pri_code
rdd_yes = .F.
sendto = '203J READY SUPPLY STORES'
supno = '0000'
route = 'Z9S'
priority = '15'
pri_code = 'C'
do get_cat
money = 'MONEY VALUE ONLY'
public signal,fund
signal = 'D'
fund = '2S'
do prt1348a
select D
use
use &item index &reqno_i,&pono_i
append blank
replace descrip with M->money
replace quant with 1
replace rcvd with .T.
use
do file_req

```


return to master

```

*****
*
* Module Name:  SEL_ACNT.PRG
*
* Author: Tom Trotter
*
* Purpose: This module allows the user to access the ACCNT.dbf,
*          scrolling forward and backward until a specific account
*          is selected for use. If the desired account is not
*          already on file, the user may opt to create it by invoking
*          NEW_ACNT.PRG from within this module.
*
* This module is called by:  DIS2_REQ.PRG
*                           DIS3_REQ.PRG
*                           NEW_ORDR.PRG
*                           SHW_NRI.PRG
*
* This module calls:  NEW_ACNT.PRG
*
*****

```

```

do while .T.
  select A
  choice = 4
  clear
  set scoreboard on
  @ 2,28 SAY "SELECT AN ACCOUNT"
  @ 3,22 SAY "-----"
  @ 4,22 SAY "[ 1. By Account Name ]"
  @ 5,22 SAY "[ 2. By Account Number ]"
  @ 6,22 SAY "[ 3. Enter New Account ]"
  @ 7,22 SAY "[ 4. Return to Main Menu ]"
  @ 8,22 SAY "-----"
  @ 10,29 SAY "Your choice:"
  @ 10,43 GET choice picture "#" range 1,4
  read
  set scoreboard off
  aname = ' '
  public accno
  accno = ' '
  do case
    case choice = 1
      use
      @ 14,27 SAY "Account Name:"
      @ 14,42 GET aname
      @ 16,27 SAY "(Blank aborts process)"
      read
      if aname = ' '
        return to master
      endif
      id = trim(upper(aname))
      use &acct index &aname_a,&accno_a
    case choice = 2
      use
      @ 14,27 SAY "Account Number:"
      @ 14,44 GET accno
      @ 16,27 SAY "(Blank aborts process)"
      read
      if accno = ' '
        return to master
      endif
      id = trim(upper(accno))
      use &acct index &accno_a,&aname_a
    case choice = 3
      do new_acnt
    case choice = 4
      use
      return to master
  endcase

```

```

if choice <> 3
  find &id
  if (eof() .or. bof())
    use
    clear
    @ 5,27 SAY "ACCOUNT NOT FOUND"
    @ 7,14 SAY "If you believe this account is in the file, try"
    @ 8,14 SAY "a shorter version of either the account name or"
    @ 9,14 SAY "number (e.g. if you are unsure if the account"
    @ 10,14 SAY "number is R1234 or R1235, look for R123 instead)."
    set color to &mscolor
    wait &wmsg
    set color to &ccolor
  endif
endif
if .not. eof()
  pick = 4
  ok = .F.
  do while .not. ok
    clear
    set scoreboard on
    @ 3,22 SAY "Account Name:"
    @ 3,37 SAY aname
    @ 5,22 SAY "Account Number:"
    @ 5,39 SAY accno
    @ 7,22 SAY "Code:"
    @ 7,29 SAY code
    @ 9,22 SAY "Research Account (T/F):"
    @ 9,47 SAY research
    @ 11,22 SAY "Requisition Serial # Range - From:"
    @ 11,59 SAY loserno
    @ 12,52 SAY "To:"
    @ 12,59 SAY hiserno
    @ 14,27 SAY "Select an action:"
    @ 14,46 GET pick picture "#" range 1,5
    @ 15,22 SAY "-----"
    @ 16,22 SAY "[ 1. Use this account ]"
    @ 17,22 SAY "[ 2. Scroll to previous account ]"
    @ 18,22 SAY "[ 3. Scroll to next account ]"
    @ 19,22 SAY "[ 4. Return to last menu ]"
    @ 20,22 SAY "[ 5. Return to main menu ]"
    @ 21,22 SAY "-----"
    read
    set scoreboard off
    do case
      case pick = 1
        accno = A->accno
        return
      case pick = 2
        if .not. bof()
          skip -1
        endif
      case pick = 3
        if .not. eof()
          skip
        endif
      case pick = 4
        ok = .T.
      case pick = 5
        return to master
    endcase
  enddo
endif
enddo

```

```

*****
*
* Module Name:  SEL_SUPL.PRG
*
* Author: Tom Trotter
*
* Purpose: This module allows the user to access the SUPT.dbf,
*          scrolling forward and backward until a specific supplier
*          is selected for use.  If the desired supplier is not
*          already on file, the user may opt to enter it by invoking
*          NEW_SUPL.PRG from within this module.
*
* This module is called by:  NEW_ORDR.PRG
*                          SUP_MEMO.PRG
*
* This module calls:  NEW_SUPL.PRG
*
*****

```

```

do while .T.
  select B
  choice = 4
  clear
  set scoreboard on
  @ 2,28 SAY "SELECT A SUPPLIER"
  @ 3,22 SAY "-----"
  @ 4,22 SAY "[ 1. By Supplier Name ]"
  @ 5,22 SAY "[ 2. By Supplier Number ]"
  @ 6,22 SAY "[ 3. Enter New Supplier ]"
  @ 7,22 SAY "[ 4. Return to Main Menu ]"
  @ 8,22 SAY "-----"
  @ 11,29 SAY "Your choice:"
  @ 11,43 GET choice picture "#" range 1,4
  read
  set scoreboard off
  public supno
  sname = ' '
  supno = ' '
  do case
    case choice = 1
      use
      @ 14,27 SAY "Supplier Name:"
      @ 14,42 GET sname
      @ 16,27 SAY "(Blank aborts process)"
      read
      if sname = ' '
        return to master
      endif
      id = trim(upper(sname))
      use &sup index &sname_s
    case choice = 2
      use
      @ 14,27 SAY "Supplier Number:"
      @ 14,44 GET supno
      @ 16,27 SAY "(Blank aborts process)"
      read
      if supno = ' '
        return to master
      endif
      id = trim(upper(supno))
      use &sup index &supno_s
    case choice = 3
      do new_supl
    case choice = 4
      return to master
  endcase
  if choice <> 3
    find &id
    if (eof() .or. bof())

```

```

use
clear
@ 5,27 SAY "SUPPLIER NOT FOUND"
@ 7,14 SAY "If you believe this supplier is in the file, try"
@ 8,14 SAY "a shorter version of either the supplier name or"
@ 9,14 SAY "number (e.g. if you are unsure if the supplier"
@ 10,14 SAY "name is Grey Electric or Gray Electic, try"
@ 11,14 SAY "looking for 'Gr' instead)."
set color to &mscolor
wait &wmsg
set color to &ccolor
endif
endif
if .not. eof()
pick = 4
ok = .F.
do while .not. ok
clear
set scoreboard on
@ 2,29 SAY "SELECTED SUPPLIER"
@ 5,25 SAY sname
@ 6,25 SAY addr
@ 7,25 SAY city
@ 7,41 SAY state
@ 7,48 SAY zip
@ 9,25 SAY fone
@ 9,48 SAY "Supplier #:"
@ 9,61 SAY supno
@ 14,27 SAY "Select an action:"
@ 14,46 GET pick picture "#" range 1,5
@ 15,22 SAY "-----"
@ 16,22 SAY "[ 1. Use this supplier ]"
@ 17,22 SAY "[ 2. Scroll to previous supplier ]"
@ 18,22 SAY "[ 3. Scroll to next supplier ]"
@ 19,22 SAY "[ 4. Return to last menu ]"
@ 20,22 SAY "[ 5. Return to main menu ]"
@ 21,22 SAY "-----"
read
set scoreboard off
do case
case pick = 1
supno = B->supno
return
case pick = 2
if .not. bof()
skip -1
endif
case pick = 3
if .not. eof()
skip
endif
case pick = 4
ok = .T.
case pick = 5
return to master
endcase
enddo
endif
enddo

```

```

*****
*
* Module Name:  SHW_NRI.PRG
*
* Author:  Tom Trotter
*
* Purpose:  This module displays items which are on file in the ITEMS.dbf
*           shown as not yet received (RCVD field = .F.).  The user may
*           direct output to the screen or printer.  He has the option of
*           listing all items marked as not received, or he may limit the
*           listing to items from a specific account.  This module may be
*           easily expanded to accomodate screens based upon date of
*           order.
*
* This module is called by:  MAIN.PRG
*
* This module calls:  WARN_PRT.PRG
*                   SEL_ACNT.PRG
*
*****

```

```

ans = ' '
clear
@ 7,28 SAY "LIST ITEMS NOT RECEIVED"
@ 8,28 SAY "-----"
@ 10,13 SAY "This program generates a listing of items ordered but"
@ 11,13 SAY "not yet received.  The list may be sent to the screen"
@ 12,13 SAY "or printer at the user's option.  The search for items"
@ 13,13 SAY "not received may also be limited within parameters"
@ 14,13 SAY "provided by the user."
@ 17,29 SAY "Continue (Y/N)?"
@ 17,46 GET ans
read
if upper(ans) <> 'Y'
    return
endif
pick = 3
clear
set scoreboard on
@ 5,25 SAY "Select Search Type:"
@ 5,46 GET pick picture "#" range 1,3
@ 6,12 SAY "-----"
@ 7,12 SAY "[ 1.  No limits (show all not received items)  ]"
@ 8,12 SAY "[ 2.  Limit list to specific account number    ]"
@ 9,12 SAY "[ 3.  Return to main menu                        ]"
@ 10,12 SAY "-----"
read
if pick = 3
    return
endif
outpt = 1
@ 13,29 SAY "Select Display:"
@ 13,46 GET outpt picture "#" range 1,2
@ 14,28 SAY "-----"
@ 15,28 SAY "[ 1.  Screen      ]"
@ 16,28 SAY "[ 2.  Printer     ]"
@ 17,28 SAY "-----"
read
set scoreboard off
counter = 0
marker = 0
value = 0.00
totval = 0.00
totcnt = 0
do case
    case pick = 1
        if outpt = 2
            prtout = .T.
            do warn_prt

```



```

clear
@ 10,31 SAY "LOAD PRINTER"
@ 12,37 SAY "&"
@ 14,25 SAY "PRESS ANY KEY TO CONTINUE"
set device to print
@ 1,0 SAY ' '
wait ''
else
prtout = .F.
endif
clear
@ 2,35 SAY trim(c_dept) + ' Dept'
@ 3,21 SAY "ITEMS NOT RECEIVED AS OF"
@ 3,47 SAY today
@ 4,1 SAY " # Item Req # "
@ 4,57 SAY " P.O. # Quantity"
@ 5,1 SAY "-----"
@ 5,56 SAY "-----"
select D
use &item index &reqno_i
go top
do while .not. eof()
if .not. rcvd
counter = counter + 1
marker = marker + 1
value = D->quant * D->uprice
totval = totval + value
totcnt = totcnt + D->quant
@ marker+5,1 SAY counter picture "###"
@ marker+5,6 SAY D->descrip
@ marker+5,45 SAY D->reqno
@ marker+5,58 SAY D->pono
@ marker+5,69 SAY D->quant picture "#####"
if marker > 14 .and. .not. prtout
marker = 1
wait &wmsg
@ 6,0 clear
endif
endif
skip
enddo
if counter = 0
@ 10,21 SAY "ALL ITEMS ARE MARKED AS RECEIVED"
else
@ marker+8,15 SAY "Total items:"
@ marker+8,28 SAY totcnt picture "#####"
@ marker+8,45 SAY "Total value: $"
@ marker+8,59 SAY totval picture "#,###,###.###"
endif
if prtout
@ marker+8,70 SAY chr(12)
set device to screen
endif
case pick = 2
do sel_acnt
key = A->accno
if outpt = 2
prtout = .T.
do warn_prt
clear
@ 10,31 SAY "LOAD PRINTER"
@ 12,37 SAY "&"
@ 14,25 SAY "PRESS ANY KEY TO CONTINUE"
set device to print
@ 1,0 SAY ' '
wait ''
else
prtout = .F.
endif
clear

```

```

@ 1,28 SAY "For Account #:"
@ 1,43 SAY key
@ 2,21 SAY "ITEMS NOT RECEIVED AS OF"
@ 2,47 SAY today
@ 4,1 SAY " # Item Req # "
@ 4,57 SAY " P.O. # Quantity"
@ 5,1 SAY "-----"
@ 5,56 SAY "-----"
select C
use &req index &accno_r
find &key
if eof()
@ 8,10 SAY "No requisitions found for account name:"
@ 8,50 SAY A->name
else
do while C->accno = key .and. .not. eof ()
key2 = C->reqno
select D
use &item index &reqno_i
find &key2
do while D->reqno = key2 .and. .not. eof()
if .not. rcvd
counter = counter + 1
marker = marker + 1
value = D->quant * D->uprice
totval = totval + value
totcnt = totcnt + D->quant
@ marker+5,1 SAY counter picture "###"
@ marker+5,6 SAY D->descrip
@ marker+5,45 SAY D->reqno
@ marker+5,58 SAY D->pono
@ marker+5,69 SAY D->quant picture "#####"
if marker > 14 .and. .not. prtout
marker = 1
wait &msg
@ 6,0 clear
endif
endif
skip
enddo
select C
skip
enddo
if counter = 0
@ 10,21 SAY "ALL ITEMS ARE MARKED AS RECEIVED"
else
@ marker+8,15 SAY "Total items:"
@ marker+8,28 SAY totcnt picture "#####"
@ marker+8,45 SAY "Total value: $"
@ marker+8,59 SAY totval picture "#,###,###.##"
endif
endif
if prtout
@ marker+8,70 SAY chr(12)
set device to screen
else
wait &wmsg
endif
endcase

```

```

*****
*
* Module Name:  STK_ORDR.PRG
*
* Author: Tom Trotter
*
* Purpose:  This module offers the user a menu option of ordering a
*           standard stock item from either NPS RSS, NSC Oakland, or
*           GSA. It subsequently invokes various modules to gather
*           the data required for the order.
*
* This module is called by:  MAIN.PRG
*
* This module calls:  GET_RDD.PRG
*                   GET_PRI.PRG
*                   GET_CAT.PRG
*                   BUY1_ITM.PRG
*                   PRT1348a.PRG
*                   FILE_ITM.PRG
*                   FILE_REQ.PRG
*
*****

```

```

choice = 4
clear
set scoreboard on
@ 4,25 SAY "STANDARD STOCK REQUISITION"
@ 6,14 SAY "This program prepares a requisition for either a"
@ 7,14 SAY "GSA or DOD standard stock item. If you want to"
@ 8,14 SAY "order forms or publications, return to the previous"
@ 9,14 SAY "menu."
@ 12,21 SAY "-----"
@ 13,21 SAY "[ 1.  NPS RSS Order          ]"
@ 14,21 SAY "[ 2.  NSC Oakland Order       ]"
@ 15,21 SAY "[ 3.  GSA Order              ]"
@ 16,21 SAY "[ 4.  Return to previous menu ]"
@ 17,21 SAY "-----"
@ 18,29 SAY "Select:"
@ 18,38 GET choice picture "*" range 1,4
read
set scoreboard off
public supno,sendto,route
do case
  case choice = 1
    sendto = '203J Ready Supply Stores'
    route = 'Z9S'
    supno = '0000'
  case choice = 1
    sendto = '(00228) NSC Oakland, CA'
    route = 'NOZ'
    supno = '1111'
  case choice = 2
    sendto = 'GSA'
    route = 'GSA'
    supno = '2222'
  case choice = 3
    return
endcase
do get_rdd
do get_pri
do get_cat
do buy1_itm
public docid,signal,fund
docid = 'A0A'
signal = 'D'
fund = "2S"
do prt1348a
do file_req
do file_itm

```

return to master

```

*****
*
* Module Name: SUP_MEMO
*
* Author: Tom Trotter
*
* Purpose: This module is conditionally invoked if the total value of
*          an order exceeds $999.99. It uses SEL_SUPL.PRG to specify
*          suppliers to include in a supplemental memo prepared to
*          accompany the requisition. Three suppliers and 2 dept
*          points of contact (POC's) must be specified. After the
*          required data is collected, it invokes PRNT_SMO.PRG to
*          print the actual memo.
*
* This module is called by: OPN_ORDR.PRG
*
* This module calls: PRNT_SMO.PRG
*                   SEL_SUPL.PRG
*
*****

```

```

clear
@ 4,24 SAY "ADDITIONAL SUPPLY SOURCES MEMO"
@ 5,24 SAY "-----"
@ 7,12 SAY "Since the total value of this order exceeds $1,000.00,"
@ 8,12 SAY "a memo, citing 2 additional supply sources, must"
@ 9,12 SAY "accompany the requisition. Two points of contact"
@ 10,12 SAY "knowledgeable about the request requirements should also"
@ 11,12 SAY "shown on the memo."
@ 14,26 SAY "Press any key to continue..."
wait ""
clear
select B
@ 3,11 SAY "This requisition shown the following company as the"
@ 4,11 SAY "primary supplier."
@ 7,20 SAY sname
@ 7,47 SAY areacode
@ 7,57 SAY fone
@ 8,20 SAY addr
@ 9,20 SAY city
@ 9,45 SAY state
@ 9,52 SAY zip
public sname1,areacode1,fone1,addr1,city1,state1,zip1
sname1 = B->sname
areacode1 = B->areacode
fone1 = B->fone
addr1 = B->addr
city1 = B->city
state1 = B->state
zip1 = B->zip
@ 13,11 SAY "You will now be requested to identify 2 additional"
@ 14,11 SAY "suppliers."
@ 16,21 SAY "Press any key to continue..."
wait ""
do sel_supl
public sname2,areacode2,fone2,addr2,city2,state2,zip2
sname2 = B->sname
areacode2 = B->areacode
fone2 = B->fone
addr2 = B->addr
city2 = B->city
state2 = B->state
zip2 = B->zip
clear
@ 12,19 SAY "Time to select one more supplier."
@ 16,21 SAY "Press any key to continue..."
wait ""
do sel_supl
public sname3,areacode3,fone3,addr3,city3,state3,zip3

```

```

sname3 = B->sname
areacode3 = B->areacode
fone3 = B->fone
addr3 = B->addr
city3 = B->city
state3 = B->state
zip3 = B->zip
clear
ans = ' '
public poc1,ext1,poc2,ext2
poc1 = ' '
poc2 = ' '
ext1 = ' '
ext2 = ' '
do while upper(ans) <> 'Y'
  @ 5,28 SAY "ENTER POINTS OF CONTACT"
  @ 6,28 SAY "-----"
  @ 9,18 SAY "Name:"
  @ 9,25 GET poc1
  @ 9,48 SAY "Ext:"
  @ 9,54 GET ext1
  @ 11,18 SAY "Name:"
  @ 11,25 GET poc2
  @ 11,48 SAY "Ext:"
  @ 11,54 GET ext2
  @ 14,30 SAY "Correct (Y/N)?"
  @ 14,46 GET ans
  read
enddo
do warn_prt
clear
@ 10,31 SAY "LOAD PRINTER"
@ 12,37 SAY "&"
@ 14,25 SAY "PRESS ANY KEY TO CONTINUE"
set device to print
@ 1,1 SAY ' '
wait ' '
do prnt_smo
return

```



```

*****
*
* Module Name:   WARNING.PRG
*
* Author: Tom Trotter
*
* Purpose: This module is used by BUY2_ITM.PRG to display a screen
*          warning that no item was entered and the order process
*          is about to be terminated.
*
* This module is called by:   BUY2_ITM.PRG
*
* This module calls:  N / A
*
*****

```

```

ans = ' '
do while .T.
  clear
  @ 6,31 SAY "WARNING !!"
  @ 7,31 SAY "-----"
  @ 9,14 SAY "Since no item was entered it is assumed that"
  @ 10,14 SAY "you desire to abort the entire order process!"
  @ 12,22 SAY "Do you want to abort (Y/N)?"
  @ 12,51 GET ans
  read
  if upper(ans) = 'Y'
    return to master
  else
    goahead = .F.
    return
  endif
enddo

```

```

*****
*
* Module Name:  WARN_PRT.PRG
*
* Author: Tom Trotter
*
* Purpose:  This module displays a warning before any output is directed
*           to the printer.  The user is advised to prepare the printer
*           or face the consequences.
*
* This module is called by:  DIS1_REQ.PRG
*                           DIS2_REQ.PRG
*                           DIS3_REQ.PRG
*                           PRT1348a.PRG
*                           PRT1348o.PRG
*                           PRT1348x.PRG
*                           PRTSF36.PRG
*                           SHW_NRI.PRG
*                           SUP_MEMO.PRG
*
*****

```

```

do while .T.
  ans = ' '
  clear
  @ 5,30 SAY "PREPARE PRINTER"
  @ 6,30 SAY "-----"
  @ 8,13 SAY "Since output will now be sent to the"
  @ 9,13 SAY "printer, it should be connected and have the power"
  @ 10,13 SAY "turned on now."
  @ 12,13 SAY "WARNING - Failure to prepare the printer may cause"
  @ 13,23 SAY "the computer to hang and possibly result"
  @ 14,23 SAY "in loss of data."
  @ 17,29 SAY "Continue (Y/N)?"
  @ 17,46 GET ans
  read
  if upper(ans) = 'N'
    return to master
  endif
  if upper(ans) = 'Y'
    return
  else
    return
  endif
enddo

```

```

*****
*
* Module Name:  WELCOME.PRG
*
* Author:  Tom Trotter
*
* Purpose:  This module displays the intital welcome banner and
*           describes the current disk configuration
*
* This module is called by:  MAIN.PRG
*
* This module calls:  N / A
*
*****

```

```

clear
@ 3,16 SAY "WELCOME TO THE NPS SUPPLY REQUISITION DATABASE"
@ 6,8 SAY "This program is designed to originate requisition documents"
@ 7,8 SAY "(DD-1348s and SF-36s), as well as maintain a database of"
@ 8,8 SAY "requisition status for a particular department. The system"
@ 9,8 SAY "is designed for use by any personnel who are familiar with"
@ 10,8 SAY "Naval Supply terminology, however, a detailed understanding"
@ 11,8 SAY "of the decision matrix used in the generation of requisition"
@ 12,8 SAY "documents is unnecessary."
@ 16,8 SAY "WARNING - If the user fails to adhere to any warnings"
@ 16,63 SAY "presented"
@ 17,19 SAY "by the program, he does so at the risk of data loss."
@ 19,19 SAY "Disks should NEVER be removed from the drives until"
@ 20,19 SAY "the program has been exited."
?
?
wait '                                Press a key to continue...'
restore from config.var
clear
@ 4,30 SAY "LOAD DISK DRIVES"
@ 5,30 SAY "-----"
@ 7,27 SAY "Database Disk Location"
if c_sdrv = c_adrv
    @ 10,20 SAY "Accounts/Supplier Disk:  ° ' Drive"
    @ 10,46 SAY c_adrv
else
    @ 9,27 SAY "Suppliers:      ° ' Drive"
    @ 9,41 SAY c_sdrv
    @ 11,27 SAY "Accounts:      ° ' Drive"
    @ 11,41 SAY c_adrv
endif
@ 16,9 SAY "Ensure that the disk drives are properly loaded before"
@ 16,64 SAY "proceeding!"
?
?
wait '                                Press a key to continue...'
return

```

LIST OF REFERENCES

1. Naval Postgraduate School, Supply Department, Customer Service Manual.
2. Naval Postgraduate School Notice 4235, "Management and Financial Control Procedures for NPS Fiscal Year 1986 Operating Funds," 26 September 1985.
3. Supply Department Training Seminar held at Naval Postgraduate School, CA, 12 November 1985.
4. Interview with LT. D.W. Olson, SC, USN, NPS Supply Control Officer, 10 January 1986.
5. Ready Supply Catalog, Naval Postgraduate School.
6. MILSTRIP Desk Guide, Publication 409, Naval Supply Systems Command, 1 March 1985.
7. Kent, Wm., "A Simple Guide to Five Normal Forms in Relational Database Theory," Communications of the ACM, v. 26, pp. 120-125, February, 1983.
8. Interview with Ms. Mary Garza, NPS Supply Department Control Division Supervisor, 26 November 1985.
9. Boehm, B.W., Software Engineering Economics, p. 18, Prentice Hall, Inc., 1981.
10. Interview with Ms. Terry Swecker, NAS Miramar Supply Automation Office, 15 November 1985.
11. Interview with Ms. Shirley Strickland, Procurement Supervisor, NSC Oakland, 8 November 1985.
12. Ashton-Tate, Inc., Advanced Programmer's Guide, 1985.

BIBLIOGRAPHY

Biggs, C.L., Birks, E.G., and Atkins, W., Managing the Systems Development Process, Prentice-Hall, Inc., 1980.

Buselt, E.J., Software Requirements Specifications of a Proposed Plant Property Management System, M.S. Thesis, Naval Postgraduate School, Monterey, California, 1983.

Carter, J.B., Software Requirements Specifications of a Proposed Plant Property Management Information System for the Naval Postgraduate School, M.S. Thesis, Naval Postgraduate School, Monterey, California, 1982.

Cunningham, V.E., A Management Information System for the Purchasing Activity at the Naval Postgraduate School, M.S. Thesis, Naval Postgraduate School, Monterey, California, 1982.

Hussain, Donna, and Hussain, K.M., Information Processing Systems for Management Richard D. Irwin, Inc., 1981.

Radford, K. J., Strategic Planning: An Analytical Approach, Reston Publishing Co., Inc., 1980.

Zychowski, T.F., A Computer-based Information System for the Issue/Receipt Control Branch, M.S. Thesis, Naval Postgraduate School, Monterey, California, 1982.

INITIAL DISTRIBUTION LIST

	No.	Copies
1. Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145		2
2. Defense Logistics Studies Information Exchange U.S. Army Logistics Management Center Fort Lee, Virginia 23801		2
3. Library, Code 0142 Naval Postgraduate School Monterey, CA 93943		2
4. Computer Technology Code 37 Naval Postgraduate School Monterey, CA 93943		1
5. LT Karen Dubay, Code 006 Information Resources Manager Naval Postgraduate School Monterey, CA 93943		1
6. Supply Officer, Code 42 Naval Postgraduate School Monterey, CA 93943		1
7. Supply Control Division Officer, Code 421 Naval Postgraduate School Monterey, CA 93943		1
8. CAPT Chuck Gibfried, SC, USN Commanding Officer Naval Supply Center Oakland, CA 94625		1
9. CDR Dana Fuller, SC, USN Commander Navy Supply System Command Washington, DC 20376-5000		1
10. LCDR Ted Case, SC, USN Commander Navy Supply System Command Washington, DC 20376-5000		1
11. Dr. Norman F. Schneidewind, Code 54Ss Administrative Sciences Dept Naval Postgraduate School Monterey, CA 93943		1
12. LCOL David E. Melchar, Code 54 Mq Administrative Sciences Dept Naval Postgraduate School Monterey, CA 93943		1
13. LCDR Hartwell T. Trotter, Jr, USN FAIRECONRON FOUR (VQ-4) NAS Patuxent River, MD 20670		2

14. LT Gerald W. Geletzke, USN
FAIRECONRON FOUR (VQ-4)
NAS Patuxent River, MD 20670

1

Thesis

T8142

c.1

Trotter

NPS Supply Requisition Database - interactive software as an alternative to written instructions.

6 NOV 87
15 JUN 89

217621

13939
35586

Thesis
T8142
c.1

Trotter

NPS Supply Requisition Database - interactive software as an alternative to written instructions.

217621



thesT8142
NPS Supply Requisition Database - intera



3 2768 000 66080 7
DUDLEY KNOX LIBRARY